IBM Z System Automation 4.4

User's Guide

SC34-2718-04

Note

Before using this information and the product it supports, be sure to read the general information under Appendix F, “Notices,” on page 237.

Editions

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

This edition replaces SC34-2718-03.

Contents

Tables

How to send your comments to IBM

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

If you have feedback to the manuals

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

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

If you have a technical problem

Use one of the following feedback methods:

- Contact your IBM service representative
- Call IBM technical support

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

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

About this publication

This publication describes IBM Z® System Automation (SA z/OS) and contains information on how to use SA z/OS to monitor and control your systems.

This publication provides an overview of the operator tasks. It points you to ways to manage your systems. For further detail, refer to the help panels or online documentation available on both 3270 panels and workstation windows.

Who Should Use This Publication

This publication is intended primarily for operators, but may also be useful for others. The information provided is specifically about using SA z/OS and assumes the following knowledge and experience:

- An understanding of z/OS
- An understanding of basic network concepts and terminology
- Using NetView
- Responding to messages and some troubleshooting

System administrators, programmers and help desk personnel may find the information helpful for installation, maintenance, and investigating and correcting problems. It also helps educate users about the SA z/OS functions.

Where to Find More Information

Z System Automation Library

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

Table 1. Z System Automation library

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

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

Related Product Information

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

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

Related Online Information

SA z/OS provides online help for all parts of its operator interfaces. From any terminal display, press PF1 to obtain help for that display.

SA z/OS also provides online help panels to support the tasks of installation and customization.

Summary of Changes for SC34-2718-03

This document contains information previously presented in IBM Z System Automation 4.2 User's Guide, SC34-2718-02.

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

The following sections list important or major changes to this document.

New and Changed Information

- (OA66029) The steps are added for how to turn a dynamic resource to a static resource. See "Dynamic Resources" on page 92.
- A description is added to introduce the recovery mode of Move and Server application groups. See "Group Recovery Mode" on page 51.
- You can now put an ISQET32 session in the new HOLD mode that keeps the session alive, as an alternative to the existing SUSPENDED mode that closes an active session. When a session is resumed, the previous filters are activated again. For more information about the HOLD mode and how to hold and resume a session, see "Using the ProcOps HOLD Session Mode" on page 234.

- System Recovery Boost (SRB) indicators are added to the ProcOps data model, API, and UIs. In addition to the ISQCCMD GETIBOSST STATUS and EVALUATE options, the ISQVARS target attention (TATTN) keyword new value "BOOST" now informs automation applications and operators when ProcOps monitoring detects system running in LPARs with BOOST mode on. For more information about how System Recovery Support is supported by Z System Automation, see “IBM Z System Recovery Boost” on page 182.
- This section “IBM Z and Logical Partitions” on page 180 is revised to describe how to perform some typical hardware operation and management tasks using the ISQCCMD common commands.

Deleted Information

- LPAR Management related information was deleted from this book, as it's no longer supported since this release.

Part 1. Introduction and Concepts

This part provides an overview of SA z/OS and its concepts. It contains the following chapters:

Chapter 1. Introducing Z System Automation

Z System Automation (SA z/OS) is a NetView-based application designed to provide a single point of control for a full range of systems management functions.

SA z/OS plays a key role in supplying high-end automation solutions. SA z/OS functions include monitoring, controlling and automating a large range of system elements spanning both the hardware and software resources of your enterprise.

Overview

SA z/OS is a systems management program with a single point of control. You see a single system image for a full range of essential systems management functions.

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 System z® or 390-CMOS zSeries 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:

- 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, or monitor them from Tivoli Netcool OMNIbus.

Component Description

System Operations (SysOps)

System Operations (or SysOps in short) monitors and controls operating system components and middleware such as CICS®, IMS, and Db2®.

With System Operations, you can automate Parallel Sysplex applications. SA z/OS can automate applications distributed over a sysplex by virtually removing system boundaries for automation through its automation manager/automation agent design. SA z/OS reduces the complexity of managing a Parallel Sysplex through its goal driven automation and its concepts, such as grouping and powerful dependency support, which enable you to model your configuration. Single systems are also fully supported; the automation scope is then just one system.

Enterprise monitoring is used by SA z/OS to update the resource status and health status information that is displayed on the Service Management Unite dashboards.

Processor Operations (ProcOps)

Processor operations (or ProcOps in short) monitors and controls processor hardware operations. It provides a connection from a focal point processor to a target processor.

With NetView on the focal point system, processor operations automates operator and system consoles for monitoring and recovering target processors.

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

Hardware Overview

Figure 1 on page 4 illustrates a basic hardware configuration.

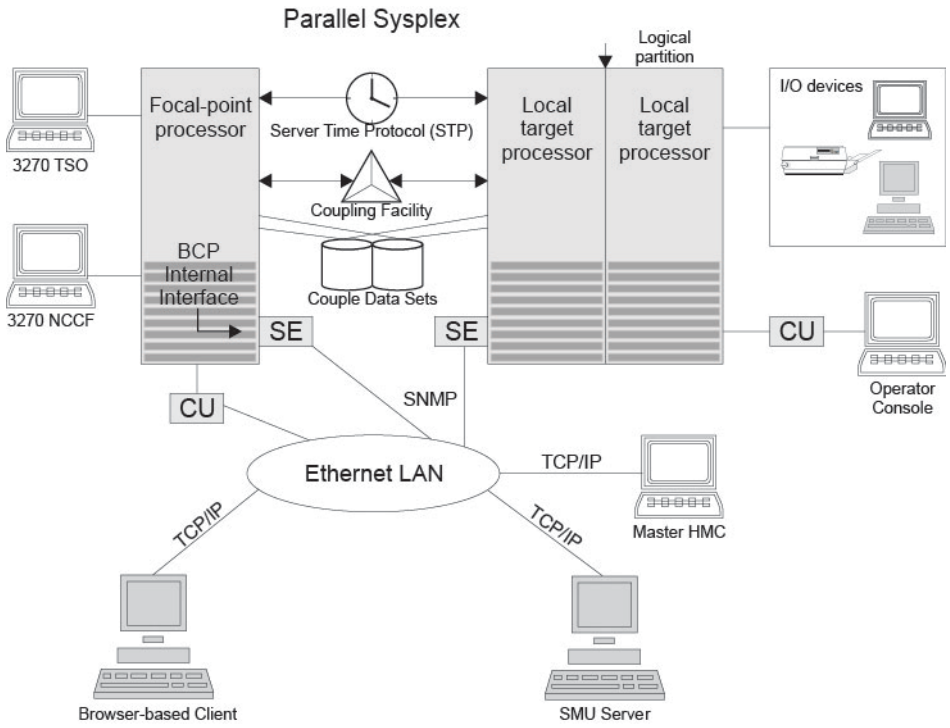


Figure 1. A Basic Hardware Configuration

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

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

and controlled using the 3270 Network Communications Control Facility (NCCF) based SA z/OS operator interfaces.

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

SA z/OS Resources

This section describes the functions and resources used by SA z/OS and its components.

Focal point processor
In a multisystem environment, the primary (or *issuing*) host processor has SA z/OS installed. You might think of it as SA z/OS's *command center*. This is where you enter commands you want to process. It includes a central processor complex, which is a collection of hardware that consists of central storage, one or more central processors, timers, and channels. NetView and SA z/OS are installed on this processor.

Target processor
The processor that is controlled by a focal point processor. It can be a local or remote processor, or a logical partition of a processor. When you send a command from a focal point system, the target is the processor you want the command to affect.

The operating system can be z/OS, OS/390, VM, VSE, Linux, or a coupling facility. If the operating system is z/OS or Linux, SA z/OS should be installed on the target processor for full SA z/OS functionality. If the operating system is VM or VSE, or if the target processor is a coupling facility, only SA z/OS processor operations functions are available for the target.

OCF-based processor
CMOS-S/390, System z operations command facility (OCF) for interacting with human operators or external programs to perform operations management functions on the central processor. Processor operations uses this facility to manage and control the processors.

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

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

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

Images
Processors can operate as *single images* or *partitioned images*. A single image allows a multiprocessor system to function as one central processor complex.

A multiprocessor system that functions as two or more independent processors (with separate power, water, and maintenance boundaries) is a *physically partitioned configuration* image.

Communications links
Paths that connect the focal point system processor to target processors so that commands, messages, and alerts can flow. Two types of communication links that can be selected for communication between the focal point and the target processor hardware: INTERNAL communication via operating system(BCPII) and hybrid SNMP connections.

Coupling Facilities
A hardware storage element with a high-speed cache, list processor, and locking functions that provides high performance random access to data for one system image or data that is shared among system images in a sysplex.

With SA z/OS system operations, you can display the status of coupling facilities from a single system's point of view or you can display sysplexwide status.

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

Automation policy
Operations and procedures (or *policy*) information is kept in files that describe routine repetitive procedures, as well as the resources in your enterprise and their relationships. Automation automatically starts, restarts, monitors, or stops resources. You define the policy using the customization dialog. The resulting file is the SA z/OS configuration file.

Processors and Systems

The SA z/OS processor operations component monitors the status of target processors (including logical partitions) and target systems. The status is determined based upon messages and alerts from the target hardware that processor operations has received at the focal point system.

Alerts are high-priority events that require operator action. Alerts are issued for situations in the target hardware and target system, and also for situations in the supporting processor operations environment. The following alerts will result in a status change of a resource, which will in turn cause a change in the representation of that resource in the SA z/OS displays, thus alerting you to a potential problem.

- Target hardware problem
- Wait state detected
- Disabled Console Communication Facility (DCCF) message detected
- Console lost
- Console access lost
- Communications task failed
- System console could not be obtained

For OCF-based processors, processor operations automates support element-generated alerts to update status of resources viewable on SA z/OS panels and graphic displays.

Messages received by processor operations at the focal point system allow processor operations to update the status of a target system as it initializes. You can view the initialization status of a target system using processor operations panels. Examples of some of the initialization statuses that can be seen are: Stage-1 Activate complete, IPL complete, IPL failed, Load failed, Waiting for IEA101A Start Message (z/OS target systems), and Waiting for VM Start Message (VM target systems).

Chapter 2. Introducing Automation Dashboards for Z Automation Web Console

Automation Dashboards for Z Automation Web Console is a customizable dashboard interface that is available with Z System Automation. It provides a single point of control for multiple SAplexes to operate in your environment.

Operators can quickly and confidently analyze, isolate, and diagnose problems by providing all relevant data including important logs in a single place.Automation Dashboards for Z Automation Web Console also enables operators to interact directly with the system by issuing commands and viewing results without going to a different console.

Automation Dashboards for Z Automation Web Console additionally allows the creation of own dashboards, providing exactly the information needed by the operations in your specific environment.

Automation Dashboards for Z Automation Web Console can be installed on Linux on z Systems or Linux on System x and uses the E2E adapter for secure communication with SA z/OS. You can download Automation Dashboards for Z Automation Web Console from IBM's download portal.

To access the download link, you need your IBM customer ID and password as well as the access key that is supplied with SA z/OS 4.3 on a CD titled ***LCD8-2753-01 Accessing IBM Z Automation Web Console Automation CD-ROM***.

Table 2. Related Documentation for Automation Dashboards for Z Automation Web Console

Related Documentation	**Location**
Automation Dashboards for Z Automation Web Console Installation and Configuration Guide	Online documentation.
Automation Dashboards for Z Automation Web Console embedded online help	Within Automation Dashboards for Z Automation Web Console, click the question mark icon (?) on a dashboard's console toolbar to get detailed information about the usage and how to customize dashboards.

Chapter 3. Concepts

This chapter contains information that you need to know while working with SA z/OS. The following topics are discussed:

Automation Concepts

This section provides details of the following automation concepts of SA z/OS:

Overview of Automation Logic

In SA z/OS, the automation function is split up, as follows:

- The observing, reacting, and doing parts are located in the NetView address space and are known as the *automation agents*. The automation agents are responsible for:
  - Recovery processing
  - Message processing
  - Active monitoring: they propagate status changes to the automation manager
- Within each sysplex, the coordination, decision making, and controlling functions are gathered into a single address space outside of NetView. This address space is called the *automation manager*.

You define the resources that you want to automate with SA z/OS using the customization dialog. The automation manager contains a model of all of the automated resources within the sysplex. The automation agents are the automation manager eyes and arms. They feed the automation manager with status information and perform the actions that the automation manager tells them to. The automation manager is the brain of automation. It does all of the decision making that involves interaction between one or more resources.

The automation manager provides *sysplexwide* automation. Its main purpose is to provide one central point of book-keeping of all resources within a sysplex, that is, the automation manager knows about the following:

- Grouping of resources
- Dependencies between resources
- Statuses of resources
- Goals for resources

The automation manager knows several different statuses for each resource:

- The observed status
- The desired status
- The automation status
- The startability status
- The compound status
- The health status

More detail about these statuses is provided in "Statuses Supplied by the Automation Manager" on page 66.

According to the available information, the automation manager makes decisions and instructs the corresponding automation agent where the resource is located to put the resource into the desired state that satisfies its goal.

The decisions are made by the automation manager with the help of *goals*. Goals can be defined either permanently by the automation administrator who creates an automation policy using the customization dialog (see *IBM Z System Automation Defining Automation Policy*), or interactively by operators who issue commands. In either case, the automation manager is informed about the goals of a certain application or resource and tries to reach the goal of a resource with its decisions.

Automation performed by SA z/OS is thus also called *goal driven automation*.

The automation manager transforms a goal (for example, the request that a certain resource or application should be up) into an order to the corresponding automation agent where the application should run.

The automation agents therefore execute orders that come from the automation manager. While carrying out the automation, the automation agents also take information from the policy that is defined for the resources. This information is available in the automation control file on each automation agent. For example, for an order to start a resource that comes from the automation manager, the automation agents retrieve information about the appropriate startup command from the automation control file.

For enterprise monitoring, the automation manager has the task of gathering and controlling information about what resources are available, what the status of the resources is, and what status updates occur during automation.

The main commands you can use to retrieve information from the automation manager are:

- INGLIST displays detail information about one or more resources (subsystem, application group, and so on).
- INGINFO displays lots of details for an individual resource or application group.
- INGVOTE displays the requests that have been issued and are currently pending for a specified resource.
- INGSCHED displays information about the current UP and DOWN service periods for resources.
- INGGROUP displays the members of a group and their settings.
- INGRELS displays the relationships that are defined for a resource.

SA z/OS supports two different types of goals:

Desired status goals
Those goals define whether a resource should be up or down. So the resource should either be available or unavailable.

Suspend goals
Those goals define whether a resource should be automated or not. So the resource should either be suspended or resumed.

Both types of goals also take into account the dependencies of resources as defined via relations in the customization dialog. Additionally, the desired status goals also check whether a trigger is defined for a resource that determines whether the availability of a resource depends upon some external events outside SA z/OS automation.

You can issue a request or goal to the automation manager with the INGREQ or INGSUSPD command, such as a goal that the specified resource should now be available, unavailable, suspended, or resumed.

Requests that are sent to the automation manager with the INGREQ or INGSUSPD command are persistent. That is, if the automation manager terminates and is restarted later, it then remembers all requests (goals) that were valid when it terminated. The automation manager will then continue to pursue all these goals for the resources. This means that if a resource should run on a certain system, and this system fails and is restarted later, the automation manager will continue to pursue the specified goals across IPL times unless they conflict with the IPL schedule times.

Desired status requests/goals

Information about desired status goals is provided to the automation manager in the following ways:

1 OPERATOR
By the operator command INGREQ (see "How Requests Become Goals" on page 15). These requests are labeled with the originator OPERATOR(USERID) in the INGVOTE command dialog.

2 AUTOOPS
By the INGREQ command issued from automation procedures (REXX exec, CLIST). These requests are labeled with the originator AUTOOPS in the INGVOTE command dialog.

3 E2EMGR
By requests from the end-to-end automation manager. These requests are handled by the SA z/OS end-to-end automation adapter, which maps the requests to SA z/OS commands.

4 EXTERNAL
By TWS Automation. If TWS Automation interfaces with SA z/OS, TWS takes care of the startup of supporting resources required for applications that are scheduled with TWS. These requests are labeled with the originator EXTERNAL in the INGVOTE command dialog.

5 SCHEDULE
There are three different ways that schedules can be used:

Resource override
The INGSCHED operator command with a resource parameter interactively overrides a service period specifically for the specified resource (see "Resource Override" on page 133).

Schedule override
With the INGSCHED operator command, which can be used to pseudo-interactively override service periods for all resources that use this service period (see "Schedule Override" on page 133).

Automation policy
With the automation policy, as defined in the customization dialog. The automation administrator uses *service periods* for applications or application groups as an easy method to let SA z/OS automatically schedule up and down times for these resources. Thus these service periods are also referred to as *schedules*.

These requests are labeled with the originator SCHEDULE in the INGVOTE command dialog.

6 A/A
By GDPS Continuous Availability Controller.

7 Default
By any other source/method, for instance runmode requests originating from INGRUN.

8 Group
When a group propagates a request to one of its members, it increments the Priority by 1000. For nested groups, this can result in multiples of 1000s.

For most of these methods, you can specify one of the following priorities for a goal that is checked in the case of two conflicting goals:

1. LOW
2. HIGH
3. FORCE (not applicable for schedules)

Table 3 on page 12 shows the hierarchy of all possible requests. The strongest goal that you can specify for a resource is interactively to issue an INGREQ command to make a resource unavailable, such as:

```
INGREQ AGFG2/APL/AOC7 REQ=STOP PRI=FORCE
```

The weakest goal you can define for a resource is to make it available with low priority via a service period.

A request to make a resource available is in most cases preferred to a request to make the same resource unavailable.

Table 3. Hierarchy of Requests

Rank/Priority	Source	Goal
03760000 FORCE	1 Operator	Unavailable
03740000 FORCE	1 Operator	Available
03680000 FORCE	6 A/A	Available
03670000 FORCE	6 A/A	Unavailable
03660000 FORCE	2 AUTOOPS	Unavailable
03640000 FORCE	2 AUTOOPS	Available
03560000 FORCE	3 E2EMGR	Unavailable
03540000 FORCE	3 E2EMGR	Available
03460000 FORCE	4 EXTERNAL	Unavailable
03440000 FORCE	4 EXTERNAL	Available
03260000 FORCE	7 DEFAULT	Unavailable
03240000 FORCE	7 DEFAULT	Available
02740000 HIGH	1 OPERATOR	Available
02720000 HIGH	1 OPERATOR	Unavailable
02680000 HIGH	6 A/A	Available
02670000 HIGH	6 A/A	Unavailable
02640000 HIGH	2 AUTOOPS	Available
02620000 HIGH	2 AUTOOPS	Unavailable
02540000 HIGH	3 E2EMGR	Available

Table 3. Hierarchy of Requests (continued)

Rank/Priority	Source	Goal
02520000 HIGH	3 E2EMGR	Unavailable
02440000 HIGH	4 EXTERNAL	Available
02420000 HIGH	4 EXTERNAL	Unavailable
02240000 HIGH	7 DEFAULT	Available
02220000 HIGH	7 DEFAULT	Unavailable
02140000 HIGH	5 SCHEDULE	Available
02120000 HIGH	5 SCHEDULE	Unavailable
01740000 LOW	1 OPERATOR	Available
01720000 LOW	1 OPERATOR	Unavailable
01680000 LOW	6 A/A	Available
01670000 LOW	6 A/A	Unavailable
01640000 LOW	2 AUTOOPS	Available
01620000 LOW	2 AUTOOPS	Unavailable
01540000 LOW	3 E2EMGR	Available
01520000 LOW	3 E2EMGR	Unavailable
01440000 LOW	4 EXTERNAL	Available
01420000 LOW	4 EXTERNAL	Unavailable
01240000 LOW	7 DEFAULT	Available
01220000 LOW	7 DEFAULT	Unavailable
01140000 LOW	5 SCHEDULE	Available
01120000 LOW	5 SCHEDULE	Unavailable
00001000	8 GROUP	Available/Unavailable

Notes:

1. A more recent request from the *same source* will however always overwrite an earlier request to the same resource without regard to the earlier request's priority. For example an operator can overwrite a HIGH priority INGREQ request to make a resource available with a later request of only LOW priority to make the same resource unavailable.
2. Within the same method, for all priorities except FORCE, the request or goal to make a resource available has a higher priority than to make it unavailable.

   However, a FORCE UnAvailable request that is issued by an operator has the highest priority.

   This is really the only method to beat SA z/OS attempts to make a resource available if for any reason you want to have this resource down.

   You can easily override your own FORCE Unavailable request for a resource with even a LOW priority request against the same resource. Thus this implementation of priorities ensures high flexibility and a quick reaction in emergency cases.

Suspend requests/goals

Information about suspend goals is provided to the automation manager in the following way:

1 OPERATOR
By the operator command INGSUSPD (see "How Requests Become Goals" on page 15). These requests are labeled with the originator INGSUSPD(USERID) in the INGVOTE command dialog.

For this type of goal, you cannot specify a priority because suspend goals cannot be conflicting. A resource is either suspended or not. Therefore, the default priority (01220000) is used for every suspend request.

Note:

1. Because suspend goals always have the same source (INGSUSPD), a more recent suspend request will always overwrite an earlier one. Therefore, it is not possible to have two suspend requests on the same resource.
2. The suspend goals are handled completely separately from the desired status goals. They are not conflicting in any case, so that it is possible to have start/stop requests and a suspend request on the same resource. The only specialty of the suspend requests is that the automation manager does not try to reach a desired status goal as long as the resource is suspended.

Summary of terminology

Goal: There are two different types of goals for a resource: the desired status goals (available or unavailable) and the suspend goals (suspended or resumed). If no goal is explicitly specified, the default goal for a resource is the desired status goal, not the suspend goal.

The success of carrying out a goal is also influenced by the dependencies of resources (defined as relationships in the customization dialog) and by triggers (triggers are taken into account only for desired status goals).

Request: If an operator specifies a goal interactively using the INGREQ or INGSUSPD command, this is called a request. According to the two types of goals for a resource, there are four requests to achieve these: MakeAvailable, MakeUnavailable, Suspend, and Resume.

Order: An order is the result of a specified goal. An operator can specify goals interactively using the INGREQ or INGSUSPD command, or an automation administrator implicitly defines goals as automation policy using service periods (service periods only define desired status goals).

The automation manager transforms such a goal, for example, the request that a certain resource (application) should be up, into an order to the corresponding automation agent where the application should run.

Vote: Resources can have relationships to other resources. As an example, a dependent resource needs a supporting resource before it can be made available. See *IBM Z System Automation Defining Automation Policy* for information on how to define relationships. If an operator generates a request to start a dependent resource with the INGREQ command, the automation manager internally generates another request to make the supporting resource available as well. Or, alternatively with an INGSUSPD command to suspend a resource, the operator can also request the suspension of the dependent resources, which internally generates a request on these dependent resources. Such an internally generated request is called a *vote*.

It may happen that votes conflict or compete with each other. Such a scenario is illustrated in "Conflicting Relationships and Requests" on page 18. In such a case, the vote with the highest priority will win. Even if votes or requests with the same priority come into conflict, the requested action decides which request will win:

- Desired status requests

The request to make a resource available is higher than the request to make that resource unavailable. To make a resource unavailable when it has the request to make it available, you must send it a higher priority request or revoke the MakeAvailable request.

- Suspend requests

  All suspend votes have the same priority, but a suspend with a SCOPE=ALL wins over a SCOPE=ONLY, so the suspend vote with greater impact will always win. If you have two suspend votes on a resource with the same scope, it doesn't matter which one is winning, because the resource is suspended anyway.

Using the INGVOTE command, you will see all currently active requests with their priorities, may they have been generated explicitly by an operator, implicitly because of the automation policy or internally as a consequence of relationships.

Automation manager: One automation manager must exist on each sysplex and one on each standalone system. In a sysplex, you can have one or more secondary automation managers (SAM) for backup purposes. That is, if the system that the primary automation manager (PAM) runs on, abends or if the automation manager itself abends, the first secondary automation manager recognizing this, will become the new PAM and will take over all current requests from the former PAM. The automation agents in the sysplex will not notice the switch from one automation manager to the other. It is important though that all automation managers (PAM and SAMs) run on systems which are in the same time zone.

How Requests Become Goals

A request asks that one specific resource (its *target resource*) should be moved to a specific state (its *goal*). A resource may, for example, be an application that should run on a certain system.

Consider the following example:

```
INGREQ abc REQ=START
```

This specifies that SA z/OS should start the resource *abc*, that is, SA z/OS should set the desired status of this resource to AVAILABLE on a system.

SA z/OS knows the dependencies for each of the actions that can be taken against the resource through the resource's relationships with its supporting resources. Using these, it propagates the request onto those resources. These internally created requests are called *votes*. Let us have a look at a CICS scenario: It makes sense to start a Terminal Owning Region (TOR_1) only after an Application Owning Region (AOR_1) is up. So you could define TOR_1 (running on a certain system) in our example to become a dependent resource and AOR_1 to become the supporting resource. So if you request to start TOR_1, SA z/OS would generate a vote to start AOR_1.

Start requests are generally propagated from dependent resources up to supporting resources so that the supporting resource is started before the dependent resource is started. Stop requests are propagated down to dependent resources in order to stop dependent resources first. Suspend requests are also propagated down to the dependent resources. The difference with suspend requests is that neither the supporting nor the dependent resource has to be suspended before the other. The suspension is executed on each of the resources in the scope of the suspend request as soon as it is possible to suspend them (no current start or stop processing active) no matter whether the supporting or dependent resource is already suspended.

How Request Propagation Works

Desired Status Requests

If the request is a higher priority than whatever request or vote the supporting resource currently has, the new request wins and the supporting resource's goal is changed to comply with the request. The process then repeats itself for the supporting resource to its supporting resources.

Eventually the propagation will stop if either SA z/OS does not encounter another supporting resource or it encounters higher priority requests or votes. At this point, SA z/OS knows which resources are not in their goal state. SA z/OS will issue start or stop commands for those resources whose prerequisites for being started or stopped are met.

Once a resource's goal has been set, SA z/OS will continue trying to achieve that goal. This means that if the resource abends and its desired status is still AVAILABLE, SA z/OS will try to restart it. If the goal changes to UNAVAILABLE, SA z/OS will stop the resource. If a resources starts when its goal is UNAVAILABLE, SA z/OS will also stop it.

Those rules only apply if a resource is not suspended. If it is suspended and not in its desired state, SA z/OS will not try to start, stop or restart it.

Suspend Requests

For suspend requests, there are no higher priority requests, because all suspend requests have the same priority. Additionally, it is not possible to have multiple suspend requests on a resource, because each one will overwrite the one before (all have the same source). The only possible thing that can happen is that multiple suspend votes are on a resource, because it is suspended over different dependency chains. In such a case, it does not matter which suspend vote is the "winning" one, because the result stays the same: the resource is suspended.

The only exception is that suspend votes from a suspend request with a SCOPE=ALL win over those requests with a SCOPE=ONLY. This ensures that a suspend request with a higher scope definitely reaches all resources in the dependency chain, so that they do not stay unsuspended because the suspend request does not reach them.

Relationship Support

One of the most effective automation capabilities of SA z/OS is the concept of *dependency relationships*. You define relationships between *dependent* resources and *supporting* resources.

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.

Relationships can be specified 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 MAKEAVAILABLE or MAKEUNAVAILABLE. Examples for conditions are WhenAvailable or WhenDown.

A typical dependency relationship that you might define for a dependent resource, A, would be: MAKEAVAILABLE/WhenAvailable, where the condition WhenAvailable applies to a supporting resource B (see Figure 2 on page 16).

So the relationship defined in Figure 2 on page 16 would read: Only if the desired state for the dependent resource A is available, make it available as soon as the supporting resource B is available.

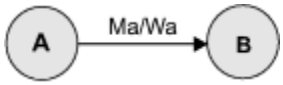


Figure 2. Relationships and Conditions

In the example shown in Figure 2 on page 16, a request to become available is issued for resource A. Resource 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 will be 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. Thus the original request is persistent until it is withdrawn.

Consider another example: assume that A has its relationship with the supporting resource B defined as HASPASSIVEPARENT, and a request to become available is issued for A. This request will not be propagated along the dependency graph. A separate request must be issued for B to make the resource A available.

There is a special relationship, HASPARENT, that simplifies relationship specifications. It is equivalent to a MAKEAVAILABLE/WhenAvailable dependency between the dependent and supporting resource and a MAKEUNAVAILABLE/WhenDown dependency from the supporting resource to the dependent resource, as shown in Figure 3 on page 17.

Figure 3. The HASPARENT Relationship

Relationships may be defined between any two resources within the same sysplex. The resources may be on different systems or may be Sysplex Application Groups. Figure 4 on page 17 presents an example of relationships across system boundaries.

Note: An application group can also be either a supporting resource or a dependent resource. It can even be simultaneously a supporting resource and a dependent resource, as shown in Figure 4 on page 17.

Be careful not to define loops.

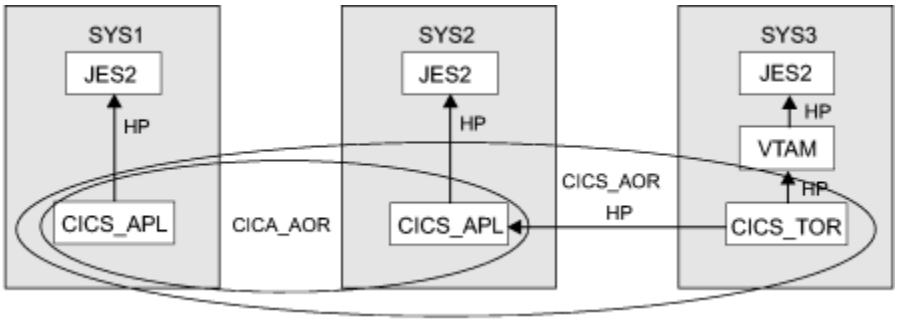


Figure 4. Relationship in a Sysplex

Figure 4 on page 17 shows an example of grouping and relationships defined across a sysplex consisting of the systems SYS1, SYS2 and SYS3.

In Figure 4 on page 17, the arrows denote HASPARENT relationships. For example, CICS_TOR will be started as soon as one member of application group CICS_AOR/APG is available and VTAM® is available.

If a group is a dependent resource, all MAKEAVAILABLE relationships apply to the members of the group, as well as to the group as a whole. For example, in Figure 5 on page 17, resource APL will not start before the resources CICS, VTAM, and JES2 are available.

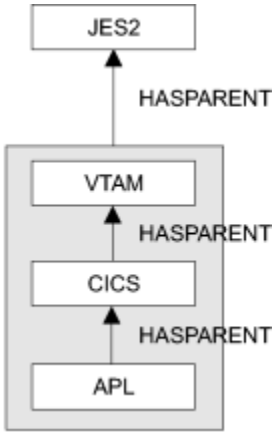


Figure 5. Relationship between Applications and Application Groups

For information on how to define relationships see "RELATIONSHIPS Policy Item" in *IBM Z System Automation Defining Automation Policy*.

Conflicting Relationships and Requests

One of the most effective automation capabilities of SA z/OS is the concept of *dependency relationships*. Using the customization dialog, the automation administrator defines relationships and conditions between dependent resources and supporting resources.

For information on how to achieve this, refer to *IBM Z System Automation Defining Automation Policy*.

In complex environments, it can happen that relationships exist that are opposed in both direction and meaning, as illustrated in Figure 6 on page 18. The following abbreviations are used in this diagram:

- For dependency relationships:

  MaWa
  denotes MAKEAVAILABLE/WhenAvailable

  MuWaD
  denotes MAKEUNAVAILABLE/WhenAssumedDown

- For statuses:

  Unav
  denotes Unavailable

  Avail
  denotes Available

- For requests or votes:

  Av
  denotes a request or vote to become available

  Un
  denotes a request or vote to become unavailable

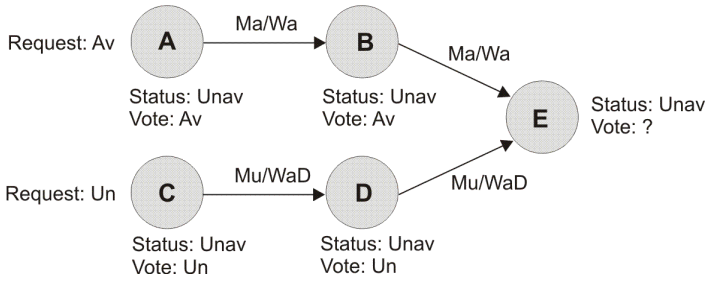


Figure 6. Conflicting Relationships and Requests

In Figure 6 on page 18, resource A is in the status Unavailable (Unav) and receives a request to become available (which internally creates the corresponding vote).

The relationship MAKEAVAILABLE/WhenAvailable propagates requests to make resources available and the relationship MAKEUNAVAILABLE/WhenAssumedDown propagates requests to make resources unavailable.

Note: The first part of a dependency relationship always applies to the resource where the arrow starts, and the condition always applies to the resource where the arrow ends.

Because requests are stored and propagated internally as votes, in Figure 6 on page 18, you see two conflicting votes that would be propagated to resource E:

- From A we have the vote to be available
- From resource C we have the vote to be unavailable

E receives both of these votes but cannot satisfy them at the same time. You can view the conflicting votes for E by issuing the INGVOTE command against this resource.

In cases when two separate streams of propagation meet, the priority of votes and requests is evaluated and used to determine the winning vote. A priority is assigned to a request either by the operator

interactively using the INGREQ command or by the automation administrator setting priorities for service periods.

A request to make a resource available wins over a request with the same priority to make it unavailable. This is only valid because of the *propagation* of requests. If, however, an operator sends a request to start a resource, and the same operator (or another one) later sends another request to stop that resource, the later request will win. The difference in this case is that this is not a *propagation* of requests.

How the Automation Manager and Automation Agents Cooperate in a Sysplex

As described in "Overview of Automation Logic" on page 9, SA z/OS automation logic in a sysplex is divided between the automation agents and the automation manager.

For reasons of high availability, SA z/OS also offers the concept of the primary and secondary automation manager (PAM and SAM), where the SAM can take over the work of the PAM. For more details, refer to the section "Manager-Agent Communication and Status Backup" in the chapter "Planning to Install SA z/OS on Host Systems" in *IBM Z System Automation Planning and Installation*.

In a sysplex, the automation agents communicate with each other using XCF. Automation managers also communicate with the automation agents using XCF and a VSAM data set for backup.

Figure 7 on page 19 shows a typical configuration in a sysplex, showing the communication paths and methods between its members.

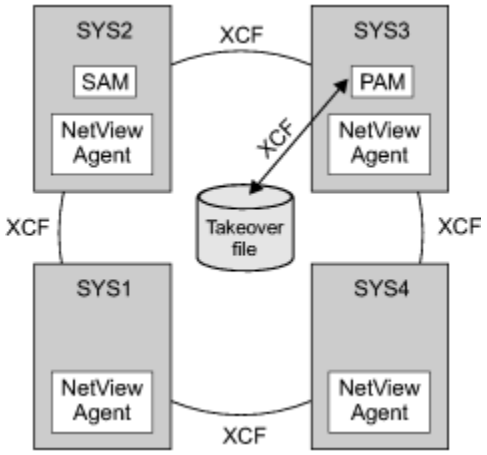


Figure 7. Automation Manager and Automation Agents Cooperation in a Sysplex using XCF

Locked Votes and Requests

When the automation manager tells an automation agent to start or stop a resource, it internally locks the current winning vote on the resource.

The automation agent is not generally expected to be able to stop an application halfway through its start sequence, or to be able to start an application if it is halfway through its stop sequence. The winning vote is locked to prevent it from changing and thus causing the automation manager to send the automation agent an order for a contrary action.

Occasionally the request that originated a locked vote will be deleted. In this case the automation manager automatically generates a copy of it (a LOCK request) that then becomes the owner of the remaining locked requests from the original vote. Votes from these requests show up on INGVOTE with a LOCK- prefix.

Once the action has been completed (or has failed) the lock on the winning vote is released, allowing it to be removed and replaced by a vote from the requests that are currently active. Once all locked votes from a locked request have been unlocked and removed, the locked request is automatically deleted.

You can use the interrupt option of the INGREQ command (that is, issue INGREQ with INTERRUPT=YES) for any start processes that are running. This forces the lock on the winning MakeAvailable vote to be released, allowing it to be replaced by your newly added request. If you use this to turn a start into a stop, it is essential that you have configured the subsystem's shutdown commands to be able to correctly shut down the subsystem from whatever state the startup may have left it in.

Note: Besides the normal start or stop requests, it can happen that a suspend request is issued, although the resource has not completed its start or stop sequence. In this case, it is also not possible to suspend the resource halfway through its processing. Because of that, the suspend request is PENDING until the resource finishes its processing (see “Reporting Suspended Resources” on page 85). But the corresponding start/stop vote is not locked, the suspend request only waits until the resource's automation status is IDLE again (see "Automation Status" on page 49).

Goal Driven Automation

SA z/OS implements *goal driven automation* in four flavors:

Flavor	Fit for what goal?
Operator requests, which can only be used during runtime: Operators can define goals for resources with INGREQ or INGSUSPD command. The goals should be active immediately after the command is issued.	• Desired status goals (available or unavailable) • Suspend goals (automated or not)
Events, triggers (see “Event and Trigger Support” on page 22), and service periods (see “Automatic Calculation of Schedules (Service Periods)” on page 21): This allows you to define desired status goals for resources (applications, application groups, or monitor resources), and to specify external events that need to be satisfied before the resources can be actually started or stopped.	• Desired status goals
Automation administrators define the default behavior or desired state of resources using the customization dialog: SA z/OS tries to keep the resource in the specified state during specified schedules under specified prerequisites.	• Desired status goals
Automation administrators can predefine suspend requests (see “Using the Suspend File” on page 89) in a suspend file, that is processed during automation manager COLD and WARM start and configuration refresh: In this way, it is possible to suspend resources right from the beginning of the manager initialization which might be useful e.g. if administrators want to define resources in the customization dialog, but those resource can not be used productively yet.	• Suspend goals

If operators want to change the goal of a resource, they may issue or remove a request to start, stop or suspend it using the INGREQ, INGSET and INGSUSPD commands with appropriate parameters. A request is executed by SA z/OS only if it does not conflict with requests of higher priority. Otherwise, because requests are persistent, they only take effect when conflicting requests of higher priority are resolved. Operators must remove obsolete requests from SA z/OS.

Note: INGREQ and INGSUSPD requests are persistent. They are remembered across session boundaries until they are explicitly revoked.

Grouping Support

Modern applications are often composed of many components, such as data servers, networking and security components. Examples are client/server applications where the application logic is distributed between a client and one or more servers and where the data can also be distributed between two and more servers. These components are often spread among the various systems in the sysplex.

In SA z/OS you can automate a complete application group. If you want to have all members (for example, all applications) of an application group available, you just tell SA z/OS to have the application group available.

Figure 8 on page 21 presents a grouping example. Complex applications that are made of individual application components are denoted with G1–G3 (G stands for group). The application components are denoted with A*n* and B*n*. For example, the complex application group G2 comprises application group G1 and the single applications A4 and A5. Application group G1 in turn consists of three applications A1, A2, and A3. Also, application groups may comprise applications that may or may not be contained in other application groups. G3 is such an example.

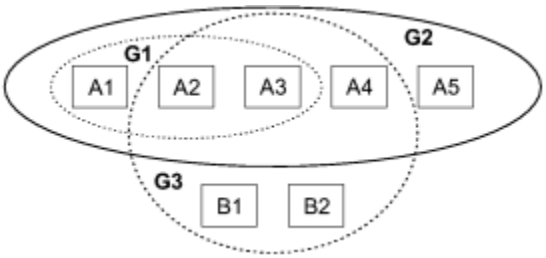


Figure 8. Example of Aggregating Applications into Application Groups

See the chapter "Scenarios on How to Use the Customization Dialog" in *IBM Z System Automation Defining Automation Policy* for a scenario that shows how grouping is performed on the conceptual level.

Automatic Calculation of Schedules (Service Periods)

Service periods allow you to schedule the availability of applications, application groups, or monitor resources. A service period is a set of service windows during which a resource (application, application group, or monitor resource) should be active or inactive.

So with service periods, you can specify both the up and down times for a resource.

Service periods set the desired status of the applications, application groups and monitor resources that they are linked to.

During the process of evaluating operator requests, SA z/OS calculates schedules for the availability of resources. This means that if you have a resource, say a database, that is supporting two applications, you can configure things so that the database will automatically be made available when either one of the two applications should be available, and to be unavailable otherwise.

If, for example, application A should be up from 7AM to 7PM, and application B should be up from 9AM to 3PM and 5PM to 9PM, simply setting these schedules for the applications and giving them a start dependency upon the database is sufficient to have SA z/OS calculate a schedule for the database of 7AM until 9PM. Note that this inference is done dynamically at runtime and will also take into account operator requests and schedule overrides.

For more details about service periods see “Using Schedules” on page 133 and *IBM Z System Automation Defining Automation Policy*.

Adaptive Scheduling

When defining the sysplex policy, the automation administrator can specify that certain resources are backups for other resources. In order to meet or maintain application availability, SA z/OS will activate these resources (if no other requests or policy definitions are opposed). This means that if an application's normal configuration is unavailable, SA z/OS will start an alternate configuration that will provide the same service.

This can be configured so that no operator intervention is required.

This feature works best when the automation administrator sets schedules on a group level, rather than on an individual resource level. Applications need to be defined in groups, because it is selective request propagation by the group resources that achieves the effect.

Event and Trigger Support

This section describes the concepts of events and triggers and how you can use them.

This is useful for understanding how to define automation policies for events and triggers using the customization dialog, see "Events Entry Type" and "Service Periods Entry Type" in *IBM Z System Automation Defining Automation Policy* for more details.

These entry types allow you to specify external conditions when certain applications on certain systems should either be automatically started or stopped.

Events are used to represent processes outside of automation. You can use the INGEVENT command to let SA z/OS know that an event has occurred. One or more events are part of a trigger condition. You can define a condition to be either a *startup condition* or a *shutdown condition*.

Events are used to decide whether the desired status goal for a resource should be realized. Even if all dependency relationships and conditions are fulfilled, but a resource's startup condition is not fulfilled, because not all events of the startup condition are set, the resource will not be started.

An event can be in one of two states: set or unset. The state of an event is changed with an INGEVENT command. For more information, see INGEVENT in *IBM Z System Automation Operator's Commands*. The state of an event is set by the user (or a procedure) and memorized throughout the sysplex. Each time, an event is set, SA z/OS checks all triggers defined in your environment or in your sysplex to find out whether applications need to be started or shut down according to the new conditions.

Triggers can be connected to applications or application groups. They act as inhibitors for the requested action. For an application to be started, its desired status must be AVAILABLE and its startup trigger must be satisfied. For an application to be stopped, its desired status must be UNAVAILABLE and its shutdown trigger must be satisfied.

If a trigger is connected to an application group then, if the trigger startup condition for the application group is not satisfied, none of its members can be started. Also, if the trigger shutdown condition is not satisfied, none of the group members can be stopped. So the trigger is kind of inherited to the members of the application group. This is also valid if the group contains nested groups. Then the trigger is also inherited to the members of the nested group.

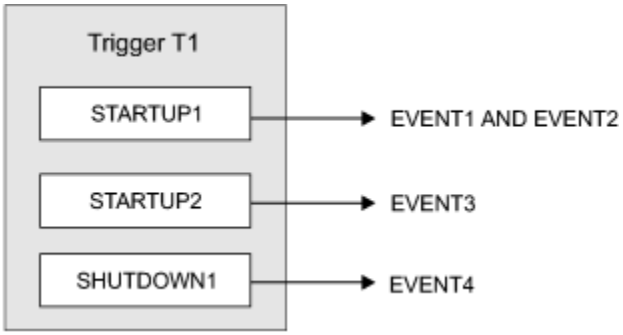


Figure 9. How a Trigger Uses Events

In Figure 9 on page 22 the application connected to Trigger T1 would be started if either EVENT1 and EVENT2 are set or if EVENT3 is set, assuming the desired goal of the application is: available. The application would be stopped, if EVENT4 is set and the desired goal of the application is: unavailable.

Triggers are used to control the starting and stopping of applications in a single system or a sysplex. So the trigger must be linked to one or more applications and has a list of conditions attached to it.

A trigger's condition list consists of up to 10 startup conditions and up to 10 shutdown conditions, each having up to *20* events. Each condition is either a startup or a shutdown condition. All events within a condition must have the status SET to meet the condition. If more than one condition of the same type, STARTUP or SHUTDOWN, is specified, *only one* of that type has to be met to satisfy the trigger.

Triggers and Traffic Lights

Like a traffic light, a trigger controls whether a running application should stop, or an application with the goal MAKEAVAILABLE should really drive away from a traffic light that switched to green, just like a ready-to start resource may only run if the trigger's startup conditions are satisfied. You see that the goal, or the request to start a resource must be there in order that a trigger with fulfilled startup conditions has an effect, just as a green traffic light is of no importance if there are no cars around the street.

However, a trigger with fulfilled shutdown conditions does not work like a red traffic light. Instead, a shutdown trigger (the red traffic light) stops the application only if the application's goal is MAKEUNAVAILABLE (while a red traffic light stops a running car anyhow). And an application with goal MAKEUNAVAILABLE is only stopped if a connected shutdown trigger is set.

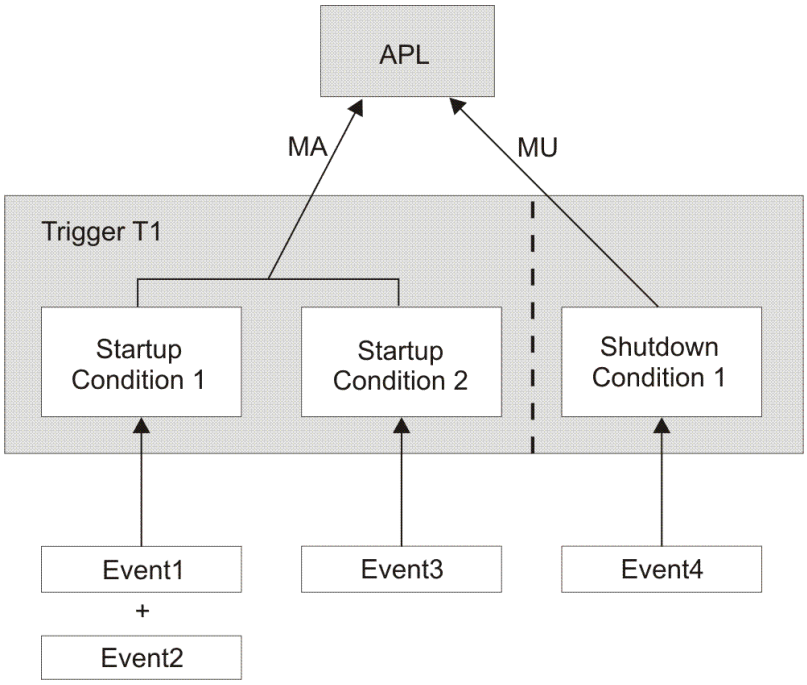


Figure 10. Triggers, Events, and Conditions

Figure 10 on page 23 shows another scenario of how triggers, events and conditions work together. For example, if EVENT1 and EVENT2 are set, Startup Condition 1 is true and Trigger T1 starts the application, provided that the application's goal is being available. If EVENT4 is set, Shutdown Condition 1 is true and Trigger T1 stops the application, provided that the application's goal is being unavailable.

A startup condition is satisfied when all events of the condition are set. A shutdown condition is satisfied when all its events are set.

Where no real dependency exists between the applications it is more appropriate to use events and triggers to control the STARTUP and SHUTDOWN.

Error Thresholds

Error thresholds influence whether or not SA z/OS recovers from an error situation. Thresholds are a number of error conditions within a certain time interval, such as five error conditions requiring restart within one hour.

During a condition that requires a restart, SA z/OS checks whether the number of occurrences of the condition reaches a defined critical threshold. If it is reached, SA z/OS does not attempt to restart the resource. You can define a critical threshold for applications to restart them.

For z/OS components, such as dump data sets or log data sets, you can define thresholds to limit how often they may be deleted after they have become full without an action being taken or a notification being sent to the operator.

Error thresholds also determine when you should be alerted to problems. The primary use of error thresholds is to track subsystem abends and ensure that the abend and restart cycle does not become an infinite loop, but you can also customize them for other uses.

The following sections show you how to define error thresholds. Information on how to display threshold settings and occurrences, and to interactively set thresholds (using the command INGTHRES) is provided in "How SA z/OS Uses Error Thresholds" on page 190.

Automation Operators

An *automation operator* is a NetView autotask, which is an operator station task (OST) that does not require a terminal or a human operator. Automation operators are crucial to SA operations. They can receive messages and issue commands for a wide variety of automation purposes.

SA z/OS sample policies, for example the *BASE add-on policy, supply definitions of the automation operators that are required to start SA z/OS. Sample automation operator profile definitions are located in the DSIOPF member, which is included in the AOFOPF* samples that are delivered with SA z/OS. You can define additional automation operators to suit your automation needs. If you modify those definitions or you do not want to use them, you need to be aware of which automation operators are required, and what messages should be assigned to them.

SA z/OS needs the definition of an *automated function* for each automation operator that it uses. One automated function can be associated with a primary and a backup automation operator (also simply called operators). These automation operators must be defined in the NetView DSIPARM data set (DSIOPF include member structure) or in the SAF product.

For example, as shown in Figure 11 on page 24, automated function NETOPER is mapped to the primary operator AUTNET1 and the backup operator AUTNET2. This two-staged definition gives you the flexibility to specify a user-specific automation operator and a second automation operator as a backup within the same automated function definition. Table 4 on page 25 shows more information about how automated functions and automation operator are mapped.

Cmd	AutoFunc	Primary Operator	Backup Operator	Messages for this Operator
___	NETOPER	AUTNET1	AUTNET2	
___	PPIOPER	AOFARCAT		
___	RECOPER	AUTREC		IEA*,IEE*,IOS*

Figure 11. Sample policy definitions of automation operators

Certain automated functions must be defined (see Table 4 on page 25). They are included in the sample policy databases that are supplied with SA z/OS. If you are not using a sample policy database, you must manually add these automated function names in your policy database. If not all required automated functions are defined, the SA z/OS initialization will fail completely.

Some automated functions are optionally required for certain functions. If not all defined, messages will come up showing that certain functions are disabled.

Mapping of Automated Functions to Automation Operators

To define the automated functions and associated automation operators (NetView autotasks), use the OPERATORS policy item of Automation Operators (AOP) entry type in the Customization Dialog.

Table 4 on page 25 lists the automated functions together with their associated automation operators that are delivered by SA, what they are used for, which automated functions are required for SA (Required=Yes), and which are required only for certain functions (Required=optional).

It is highly recommended that you assign the suggested automated operators to these required automated functions as listed in Table 4 on page 25. If you attempt to share automated operators across more than one automated function, you will experience a significant performance degradation during SA z/OS initialization or configuration refresh.

Table 4. Mapping of NetView Automation Operator to SA z/OS Automated Function

Automated Function	Automated Operator	Required	Description
INITOPR1	AUTINIT1	YES	Used for SA z/OS initialization.
INITOPR2	AUTINIT2	YES	Used for SA z/OS initialization.
BASEOPER	AUTBASE	YES	Serves as a backup for all other automation operators.
SYSOPER	AUTSYS	YES	Used as the primary SA z/OS automation operator. Especially used for messages IEE400I and IEE600I.
LOGOPER	AUTLOG	YES	Used for message writing and for unassigned messages.
MSGOPER	AUTMSG	YES	Used for message formatting and routing.
AOFWRK*nn*	AUTWRK*nn*	YES	Used for APL and monitor resource processing, for example startup, shutdown. For more information, see "Special Automated Functions AOFWRKnn" on page 26.
NETOPER	AUTNET1, AUTNET2	YES	Serves as the automation operator for network products (VTAM and NetView).
JESOPER	AUTJES	YES	Serves as the automation operator for JES2 or JES3 product.
MONOPER	AUTMON	YES	Used for active monitoring of resources.
RECOPER	AUTREC	YES	Serves as the primary recovery operator. It is not recommended to use it for user-specific automation that issues WTOs and WTORs. Otherwise, it would halt the SA z/OS WTO and WTOR buffer recovery process, which can result in a buffer full situation.
RPCOPER	AUTRPC	YES	Used for XCF requests.
SHUTOPER	AUTSHUT	YES	Serves as the base shutdown operator.
GSSOPER	AUTGSS	YES	Used for general automation and communication services for the automation manager (AM), including processing orders sent from the AM to the automation agents. For more information, see "Special Consideration of GSSOPER" on page 27.
XCFOPER	AUTXCF	Yes	Used for enhanced Parallel Sysplex® automation functions.
XCFOPER2	AUTXCF2	Yes	Used for enhanced Parallel Sysplex automation functions.
EVTOPER	AUTEVT1, AUTEVT2	Yes	Used for event subscription and the TWS command request interface.
GATOPER	GAT&DOMID.	Optional	Serves as the outbound gateway operator for automation notification message forwarding. It should not be targeted to handle user commands.
MVSCONS*i*	AUTCON	Optional	Used for the MVS and NetView console interface.
PPIOPER	AOFARCAT	Optional	ARM PPI receiver.

Table 4. Mapping of NetView Automation Operator to SA z/OS Automated Function (continued)			
Automated Function	**Automated Operator**	**Required**	**Description**
HWOPER01-nn	AUTHW001-nnn	Optional	Used for Hardware (Processors) with connection type 'INTERNAL'. The minimum number of required HWOPERnn entries is the number of defined Processors with connection type 'INTERNAL' + 1. The Automation Operator can have a different name from AUTHWnnn.
PLEXOPER	AUTPLEX	Optional	Used for enhanced Parallel Sysplex automation functions.
PLEXOPR2	AUTPLEX2	Optional	Used for enhanced Parallel Sysplex automation functions.
PLEXOPR3	AUTPLEX3	Optional	Used for enhanced Parallel Sysplex automation functions.
AOFSES*nn*	AUTSES*nn*	Optional	Used for OMEGAMON® sessions.
ALRTOPER	AUTALERT	Optional	Used for alert-based notification.
OPCACMDR	AUTOPCR	Optional	Used to execute requests from TWS batch jobs.
OPCAMSTR	AUTOPCP	Optional	Used to receive requests from TWS.
OPCAOPR2	AUTOPCE	Optional	Used to execute requests from TWS.
AOFTWS01-06	AUTTWS01-06	Optional	Used for TWS requests.
AOFRDSEV	AUTRDSEV	Optional	Used to execute requests from the RDS Archiver.
AOFRDSAR	AUTRDSAR	Optional	Used to execute requests from the RDS Archiver.
AOFCMD01-05	AUTCMD01-05	Optional	Used to execute requests from the Command Receiver.
CICSMSTR	AUTCICS	Optional	Used for CICS Automation.
IMSMSTR	AUTIMS	Optional	Used for IMS Automation.
E2EOPER	AUTE2E	Optional	Used to execute requests from the E2E Adapter.
E2EOPR01-03	AUTE2E01-03	Optional	Used to execute requests from the E2E Adapter.
TECOPER	AUTOTEC	Optional	Used for TEC Notification.

Special Automated Functions AOFWRK*nn*

SA z/OS uses special automated functions AOFWRK*nn* for application and monitor resource processing. They are also named *work operators*.

AOFWRK*nn* are used to distribute the work on different automation operators (NetView autotasks) so that they can run in parallel. These automation operators are responsible for the startup, shutdown, monitoring, and message handling activities. Additionally, they ensure that all messages for an APL are processed in the correct sequence.

The naming format of these automated functions is AOFWRKnn, where *nn* can be any number. The *nn* part of the automated function name does not have to be in sequence.

At least **three** automated functions must be defined. You can define more to fulfill how much parallelism you want. SA z/OS recommends using 20 automated functions, AOFWRK01 through AOFWRK20, with the associated automation operator names AUTWRK01 to AUTWRK20. You can increase this number according to the installation needs.

These work operators are used as dedicated automation operators responsible for all the work that is related to an application or monitor resource. However, if there are more application and monitor resources defined than work operators, several applications and monitor resources will share the same work operator.

At SA z/OS initialization or configuration refresh, each defined application and monitor resource will be assigned an automated function from the pool of defined work operators in a round-robin manner. Thus, each work operator has a list of applications and monitor resources that it is responsible for. Each work operator then subscribes the messages of those applications via the NetView ASSIGN command. Finally, the initial monitoring of SA z/OS runs on the appropriate work operator, which is then locked until SA z/OS initialization is complete (message AOF540I issued). After SA z/OS is fully initialized, all messages for an application are queued to the same work operator. It ensures that all messages are processed in the order they have been received.

Special Consideration of GSSOPER

The automated function GSSOPER is used for System Operations commands ISSUEACT, ACTIVMSG, TERMMSG, and HALTMSG. These commands (routines) are routed from the NetView Automation Table to GSSOPER. GSSOPER will no longer execute these routines, it will distribute them to the responsible AOFWRK*nn* automated functions (see “Special Automated Functions AOFWRKnn” on page 26).

GSSOPER processes start and stop orders coming from the automation manager, before they are distributed to the responsible AOFWRKnn automated function. GSSOPER will also be used as backup when there is no AOFWRK*nn* automated function defined. For more information about the System Operations commands, see *Programmer’s Reference*.

Message Automation

Message automation means acting upon the appearance of the trapped messages using action rules defined in the policy, for example, issuing commands, responding to replies, checking for certain codes within a message, processing user data, etc. SA z/OS allows you to automate messages from z/OS, applications, logs and, hardware.

SA z/OS exploits the NetView Automation Table (AT) and Message Revision Table (MRT). The AT contains traps for messages that must be automated. A related AT entry is required to call a command to execute the action. The MRT allows you to modify message attributes such as color, route code, descriptor code, display and syslog settings, and text of original z/OS messages.

Messages that pass all the filters (MPFLSTxx and MRT) end up in the AT containing action rules.

You can define ATs, MRT, and the MPFLSTxx PARMLIB member via the Customization Dialog. For example, if an action must be taken in response to a message, this action can be defined in the MESSAGES/USER DATA policy item of the Customization Dialog.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                            Message Processing       Line 00000001 Col 001 075

Entry Type : Application           PolicyDB Name   : MYPDF
Entry Name : MYAPLS                Enterprise Name : MYPDF

Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)
               I, D (insert or delete lines), CN (Class Name)
Message ID field length. . 16   (1 - 32)

Cmd Message ID       Description                              C  R  K  U A M F
___ ________________ _________________________________________
___ ________________ _________________________________________
___ ________________ _________________________________________
```

SA z/OS can generate the ATs, MRT, and the MPFLSTxx PARMLIB member by using the Customization Dialog Build function. You can define whether an AT/MRT is built on enterprise, sysplex, or system level, or skip building an AT/MRT in the BUILD CONTROL policy item of MSG entry type.

```
                          Define Build Control Parameter

PolicyDB Name : USER_PDB                Enterprise Name : USER_ENTERPRISE

AT/MRT scope . . . . . . . . SYSTEM___ (NONE ENTERPRISE SYSPLEX SYSTEM)
```

For more information, see Chapter "Automation Table, Message Revision Table, and MPFLSTxx member" in *Defining Automation Policy*.

Conceptual Overview

This section gives a brief overview of the main aspects of SA z/OS message automation:

- SA z/OS typically traps a message by its unique message ID. See “What's A Message” on page 29.
- In general, AT entries trap messages independently of the issuing job. But you can restrict the message to a certain job by specifying an adequate condition, for example, using the application entry's job name as an additional condition.
- SA z/OS provides a comprehensive set of predefined messages with related AT, MRT, and MPF definitions. You can use these predefined messages out-of-box or change them for your environment. These messages are listed under Customization Dialog > MVC entry type > +SA_PREDEFINED_MSGS entry > MESSAGES/USER DATA policy item. You can use this MESSAGES/USER DATA policy item to view what AT, MRT, or MPF entries are built by SA z/OS for automating a particular message.
- You can make the following definitions by using the MESSAGES/USER DATA policy item in the Customization Dialog:
  - Define a set of commands that are executed when a message is detected. See “Defining Actions for Messages” on page 30.
  - Define a reply that is issued as response to a message. See “Defining Actions for Messages” on page 30.
  - Assign a status for an application resource change. See “Defining Messages for Status Change (Status Messages)” on page 32.
  - Capture the message for further display and information. See “Defining Messages for Capturing” on page 33.
- SA z/OS provides predefined AT, MRT, and MPF entries for above cases. You can also overrule them by marking a message to be ignored, thus not generating an AT, MRT, or MPF entry or by specifying individual AT, MRT, and MPF specifications. All those individual specifications can also be done in the Customization Dialog.
- For size and performance reasons, AT, MRT, and MPF entries are structured in a message-oriented rather than job-oriented way.
- One single AT entry is generated for a message regardless of under what application, monitor resource or MVS component entry the message is defined. For actions, SA z/OS takes care that only the actions defined for the application, monitor resource, or MVS component entry representing the issuer of the message, are executed.
- SA z/OS provides a special concept called *pseudo messages*.
  - Pseudo messages invoke commands at the time when a resource enters a defined status, for example, UP, STARTED, STOPPING, STUCK, DOWN, etc. Those invoked commands are also called *Status Commands*.
  - Special predefined pseudo messages, like CAPMSGS, INGALERT, and others are used to implement special functions within SA z/OS.

  For more information, see “Pseudo Messages” on page 38.

What's A Message

In general, a message is composed by the first token and message text: <first-token> <message-text>.

The message ID distinguishes a message from all others and typically appears as the first token. There are some exceptions that the actual message is wrapped by the outer message and hence the unique message ID appears within the message text. For a general description of messages, see Format of the message body in *z/OS MVS System Messages Vol 1 (ABA-AOM)*.

SA requires a unique token to trap a particular message and that token is typically the message ID. Regardless of whether the message is wrapped or not, what matters to the users is the unique message ID that they would add in the policy.

SA has predefined message rules to extract the unique message IDs for different messages:

- Messages from UNIX System Services processes (See "Hints and Tips" in *IBM Z System Automation Customizing and Programming*.)
- Hardware messages from SA z/OS BCPii or Processor Operations (See "Messages Issued by a Processor Operations Target System" in *IBM Z System Automation Customizing and Programming*.)
- Job Log Monitoring messages generated by System Operations (See "Job Log Monitoring" in *IBM Z System Automation Customizing and Programming*.)

Message Processing

SA z/OS assigns all the messages of a certain application to the work operator (AUTWRK*nn* by default) that is associated with the application. If the AT action uses standard SA z/OS capabilities (that is, SA z/OS commands), the work operator processes the messages in the following steps:

1. The message flows through the ATs.
2. If the message matches the AT condition, then SA z/OS data model is applied, which includes automation flag checking, code matching, threshold comparison, pass evaluation, and message capturing.
3. Finally, the command is executed, or the outstanding reply is answered.

Advanced users

The following information is intended for advanced users.

There are two places where you can modify the processing for single messages:

- You can overrule the AUTWRK*nn* automation operators where SA assigns the messages by default. Follow these steps:

  1. First, Set the AOF_ASSIGN_JOBNAME advanced automation option to 0, which enables "ASSIGN BY MESSAGE ID" take precedence over "ASSIGN BY JOBNAME".
  2. Then, use the NetView ASSIGN command with the MSG parameter to redirect the message to the automation operator you want. That particular message is then assigned to your specified automation operator, while all other messages still run on the AUTWRK*nn* automation operators that are assigned by SA z/OS.

  However, make this modification with care because it suspends SA z/OS load balancing and breaks the serialized command processing for that application.
- You can overrule the automation operator that SA assigned to execute the command. To do so, specify an automated function together with the command in the MESSAGES/USER DATA policy in the Customization Dialog.

  Execution of the command is then routed to the automation operator that is specified for the automated function. The AT and data model processing are running on the specified automation operator and thus proper sequencing is guaranteed.

  SA z/OS internally uses the AOFEXCMD command (see *Programmer's Reference*) to queue the command to the specified automation operator. The AOFEXCMD routine checks whether the requested

automation operator is available. If not available, it queues the command to a backup operator, so that in any case the command does not run on the current automation operator.

It is recommended that you make this modification only for special reasons, for example, for long running commands. It's because it can break the serialized command processing for that application, if not all commands are executed on the same automation operator.

End of Advanced users

Defining Actions for Messages

SA z/OS allows you to define actions (commands or replies) for Application (APL), Monitor Resource (MTR), and MVS Component (MVC) policy entries.

The following information describes how to define commands or replies as actions. For example, suppose that you need to define a command action for a message XYZ222I for the HSM application, where XYZ222I is a new message that is not predefined by SA z/OS.

1. The first step is to select an entry type (APL, MTR, or MVC) on the **Entry Type Selection** panel and a specific entry, which leads to the **Policy Selection** panel. In this example, select the APL entry type and the HSM application.
2. On the **Policy Selection** panel, select **MESSAGES/USER DATA** policy item, which leads to the **Message Processing** panel. It's where you can define actions for messages.
3. Specify the ID of the message that you want to automate. You can use system automation symbols (AOCCLONE variables) or system symbols in message IDs. These symbols are resolved when AT and MRT are loaded, but cannot be resolved for MPF.

   Out of the box, SA z/OS supports trapping messages only by message IDs and a few other conditions that can be specified on the **Message Processing** panel. If these capabilities are not sufficient, you can define the AT entry yourself by defining an AT Override.

   In this example, enter the message ID XYZ222I.

```
                            Message Processing       Line 00000001 Col 001 075

Entry Type : Application           PolicyDB Name   : USER_PDB
Entry Name : MYAPL                 Enterprise Name : USER_ENTERPRISE

Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)
               I, D (insert or delete lines), CN (Class Name)
Message ID field length. . 16   (1 - 32)

Cmd Message ID       Description                                C  R  K  U A M F
 c  XYZ222I          ___________________________________________
___ ________________ ___________________________________________
```

4. Enter R (reply) or C (command) in the column next to the specified message ID, and define the specific command or reply action on the displayed **Reply Processing** or **Command Processing** panel.

   In this example, enter C and define the following command for XYZ222I.

```
            Command Processing : XYZ222I             Line 00000001 Col 001 075

Mixed case . . . NO   (YES NO)

Cmd Ps/Select  AutoFn/* Command Text
___ __________  ________ MSG OPER1 ***RETRIEVED MESSAGE XYZ222I ***
```

5. After your reply or command definition, press PF3 to return to the **Message Processing** panel. Issue A or M in the **Cmd** column next to the specified message ID, the AT, MRT, and MPF entries that SA generated based on your specification are then displayed.

```
                        Message Automation Overview

 AC AT condition     MS MRT action selection     MF MPFLSTxx specifications
 AS AT status        MO MRT override             IG Ignore for AT, MRT, MPF
 AO AT override

Message ID : XYZ22I

Generated AT entry :
IF MSGID = 'XYZ22I' THEN
 EXEC(CMD('ISSUEACT') ROUTE(ONE %AOFOPGSSOPER%));

Generated MRT entry :
UPON ( MSGID = 'XYZ22I' )
REVISE('Y' AUTOMATE)

Generated MPF entry :
XYZ22I
```

This definition leads to the creation of an AT entry for message XYZ222I using the ISSUEACT command after the next configuration build process. If you want to pass parameters to ISSUEACT, you can do so as described in “Defining AT Overrides” on page 44.

For messaged defined in the MESSAGES/USER DATA policy item for MVS Components (MVC), the parameter SYSTEMMSG=YES is added to the generated ISSUEACT command.

For more information about ISSUEACT, see *Programmer's Reference*.

Defining Code Match Table

SA z/OS allows you to define a code match table, also called "codes" (COD) for Application (APL), Monitor Resource (MTR), and MVS Component (MVC) policy entries. It allows a flexible way to specify the parameters for certain SA z/OS commands.

Code match checking is used by SA z/OS in the following situations:

- Processing of messages with ISSUEACT to trigger appropriate commands or replies with the possibility to assign also an appropriate severity for message being captured, for example, IMPORTANT. See "ISSUEACT" in *Programmer's Reference*.
- Processing of messages or OMEGAMON exceptions with INGMON to update the status of an MTR or to issue appropriate commands or replies. See "INGMON" in *Programmer's Reference*.
- Processing of termination messages with TERMMSG to determine which status to place the resource in, for example, ABENDING. See "TERMMSG" in *Programmer's Reference*.
- Processing of WTORs with OUTREP to assign a severity, for example, IMPORTANT. See "OUTREP" in *Programmer's Reference*.
- Capturing of messages with AOFCPMSG with the possibility to specify any parameter of AOFCPMSG. See "AOFCPMSG" in *Programmer's Reference*.
- Sending events with INGALERT to event notification targets such as IOM, EIF, TTT, or USR. See "Alert-Based Notification" in *Customizing and Programming*
- Processing of WTO or WTOR buffer shortage recovery. For details, see "Enabling WTO(R) Buffer Shortage Recovery" in *Customizing and Programming*.
- Processing of CICS or IMS transaction abend recovery using message ID ABCODETRAN. For details, see Chapter 3 and 10 in *Product Automation Programmer's Reference and Operator's Guide*.
- Processing of IMS BMP region abends using message ID ABCODEPROG. For details, see "ABCODEPROG: Respond to IMS Application Abends" in *Product Automation Programmer's Reference and Operator's Guide*.

Defining Messages for Status Change (Status Messages)

Status messages are those messages that indicate a status change of the resource.

Many status messages of APL, MTR, and MVC resources are known to SA z/OS. The related AT entries are already predefined. For example, for a message indicating that an APL is available, ACTIVMSG command is triggered. For a message indicating that an APL is unavailable, TERMMSG command is triggered.

For these predefined messages, there is no need to have additional definitions in the policy database.

Note: There are certain messages that can be used as status messages and that require additional code definitions, for example, IEF450I. TERMMSG sets the status depending on these definitions. For more information about TERMMSG, see *IBM Z System Automation Programmer's Reference*.

The code definitions for message IEF450I are important. Sample policy *BASE includes a set of recommended code definitions.

Display existing status messages

To display all predefined and user-defined status messages and their associated statuses, use the Status Message Report in Customization Dialog.

1. On the Customization Dialog primary panel, select option 3 (Report).
2. Select option 8 (StatusMsgs) to create a status messages report. Pressing Enter returns you to the **Report Selection** panel.
3. Select option 9 to view the report. As the following screen excerpt shows, the report shows the message ID, resource status indicated by this message, message description, the entry name, and the entry type that the message is defined for.

```
Message ID |Status      |Description                  |Entry Name          | Typ
-----------+------------+-----------------------------+--------------------+------------
CQS0020I   |UP          |CQS up message               |+SA_PREDEFINED_MSGS | MVC
TEST001I   |UP          |Test application has started | APPL1              | APL
```

Define additional status messages

If a status message is not predefined by SA z/OS, you can define it additionally for APL, MTR, and MVC policy entries. The A or M action leads to the **Message Automation Overview** panel, where you can enter the AS option to display the **AT Status Specification** panel that lists resource statues.

The following example defines an APL's UP status, which is indicated by message XYZ444I. On the **Message Processing** panel, specify the XYZ444I message ID and enter A or M in the **Cmd** field next to it.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                            Message Processing       Line 00000001 Col 001 075

 Entry Type : Application           PolicyDB Name   : DEMO
 Entry Name : MYAPL                 Enterprise Name : DEMO

 Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)
                I, D (insert or delete lines), CN (Class Name)
 Message ID field length. . 16   (1 - 32)

 Cmd Message ID       Description                              C  R  K  U A M F
  A  XYZ444I          ________________________________________
 ___ ________________ ________________________________________
```

On the displayed **Message Automation Overview** panel, enter the AS option to display the **AT Status Specification** panel and select the UP status.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                              AT Status Specification
Command ===>

View or change specifications for message: XYZ444I

Application / MVC Component status      Monitor health status
_ ACTIVE      Starting                  _ NORMAL   Resource shows good results
s UP          Available                 _ WARNING  Resource shows degradation
_ HALTED      Degraded                  _ MINOR    More severe than WARNING
_ Terminating Received STOP command     _ CRITICAL More severe than MINOR
_ Terminated  Terminated normally       _ FATAL    More severe than CRITICAL
_ ABENDING    Going to end abnormally   _ UNKNOWN  Health status not available
_ Abended     Ended abnormally          _ Check    Evaluate health status
_ BREAKING    Non-recoverable abending  Monitor status
_ BROKEN      Non-recoverable abend     _ FAILED   Monitor failed
                                        _ BROKEN   Monitor finally failed
Capture specification
_ Capture     Capture Message
              DOM . . .________________________________
```

Press PF3 to return to **Message Automation Processing** panel. This definition leads to the creation of an AT entry for message XYZ444I using the ACTIVMSG command after the next configuration build process, as shown on the **Message Automation Overview** panel.

```
  COMMANDS  VIEW  HELP
------------------------------------------------------------------------------
                         Message Automation Overview

 AC AT condition     MS MRT action selection     MF MPFLSTxx specifications
 AS AT status        MO MRT override             IG Ignore for AT, MRT, MPF
 AO AT override

Message ID : XYZ444I

Generated AT entry :
IF MSGID = 'XYZ444I' THEN
 EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ONE %AOFOPGSSOPER%));

Generated MRT entry :
UPON ( MSGID = 'XYZ444I' )
REVISE('Y' AUTOMATE)

Generated MPF entry :
XYZ444I

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

Defining Messages for Capturing

From the hundreds or thousands of messages that can be issued as WTO or WTOR, only a subset is of particular interest from the operations point of view. These are messages that either represent a specific status of a resource controlled by SA z/OS or that can trigger an automated action as specified in the automation policy. Therefore, it is necessary that the messages that represent such situations are retained and presented to the operations team. These messages have to be retained until the situation is resolved, in which case they can be deleted automatically by SA z/OS, or manually by the operator.

Furthermore, SA z/OS allows you to assign a severity to the captured messages. Based on the severity, it can relay such messages to other targets. See “Determine Where Captured Messages Are Sent” on page 37 for more information. SA z/OS distinguishes the following severities:

Severity	Meaning
IGNORE	The message will not be captured.
NORMAL	The value NORMAL is used for operator information, for instance to log the message that indicates a resource has entered the UP status. You can view NORMAL messages by the DISPINFO command for applications, DISPMTR command for monitor resources, and DISPSYS command for MVS components.

UNUSUAL (default) IMPORTANT CRITICAL	Messages with UNUSUAL, IMPORTANT, or CRITICAL severity are called *exceptional messages*, that is, messages that an operator needs to be aware of and that typically result in automated recovery actions and/or events being generated. You can display exceptional messages by the INGMSGS command and the Status Display Facility (SDF), in addition to the DISPINFO, DISPMTR, and DISPSYS commands. Exceptional messages are retained until they are deleted automatically by SA z/OS or explicitly by the operator. For details about using INGMSGS to display and delete exceptional messages, see "INGMSGS" in *IBM Z System Automation Operator's Commands*.

When a message is captured, it is held in a resource-related buffer with a limited number of entries. In addition, any exceptional message is held in a system-related buffer that is limited in size, too. The sizes of these buffers can be defined in the policy, see “Policy Definitions for Message Limits” on page 35.

Both types of buffers are organized in form of a ring so that once the limit is reached, the slot of the oldest message is taken by the most recent message added to a buffer. The buffers are not persisted, and they are released by the system when the NetView session terminates. Whenever NetView is started, these buffers start to fill anew.

Conditions for Message Capturing

SA z/OS captures all messages that matter to your automation and where SA z/OS takes responsibility.

Messages are captured in the following conditions:

- Commands are defined for the message in the policy. See “Defining Actions for Messages” on page 30.
- A reply is defined for the message in the policy. See “Defining Actions for Messages” on page 30.
- A particular resource status is selected for the message on the AT Status Specification panel, for instance an UP status. See “Defining Messages for Status Change (Status Messages)” on page 32.
- You explicitly capture the message either by selecting the Capture field in the AT Status Specification panel or by invoking the AOFCPMSG command. See “Policy Definitions for Captured Messages” on page 35.

For compatibility reasons, messages are captured even if a resource is suspended, or more generally, when its agent's automation flag is turned off. If you want to avoid message capturing when a resource is suspended, you can set the **Capture Messages** field to NO in the AUTOMATION OPTIONS Policy Item for System Defaults (SDF).

```
   COMMANDS  HELP
------------------------------------------------------------------------------
                           System Automation Options

 Entry Type : System Defaults       PolicyDB Name   : MYPDF
 Entry Name : SYSTEM_DEFAULTS       Enterprise Name : MYPDF
                                                            More:     +
 System specific definitions:
   Exceptional Messages Limit . 300       (0 to 99999999, default 300)
   IPL Complete Time Limit. . .           (00:00:00 to 24:00:00)
   IPL Complete Status  . . . .
                                          (AVAILABLE SOFTDOWN STANDBY)
                                          (or SATISFACTORY)
 Resource specific definitions (APLs, MTRs) if automation is suspended:
   Capture Messages . . . . . NO          (YES NO, default YES)
   Check Thresholds . . . . . YES         (YES NO, default YES)
   Manage WLM resources . . . . YES       (YES NO, default YES)

 Resource specific definitions (APLs, MTRs, System APGs):
   Captured Messages Limit. . . 20        (0 to 999, default 0)
   Desired Available. . . . . ALWAYS      (ALWAYS ONDEMAND ASIS, default ALWAYS)
 Command ===>_____________________________________________________________
```

Policy Definitions for Message Limits

SA z/OS allows you to limit the number of messages that can be captured for an individual automation resource (APL, MTR), MVS component defaults (MDF), system defaults (SDF), or application defaults (ADF). In absence of a custom default value, that limit is zero.

To set the captured message limit, specify a value in the **Captured Messages Limit** field in the following policy items respectively.

- To set the limit for an individual APL instance or class, use the APPLICATION INFO policy item for Applications (APL).
- To set the limit for an individual MTR, use the MONITOR INFO policy item for Monitor Resources (MTR).
- To set the default limit for resource MVSESA, use the MVSESA INFO policy item for MVSCOMP Defaults (MDF).
- To set the default limit for a system, use the AUTOMATION OPTIONS policy item for System Defaults (SDF).
- To set the default limit for all APLs, use the APPLICATION INFO policy item for Application Defaults (ADF).

Exceptional messages have their own limit that can be specified by setting the **Exceptional Messages Limit** field in the AUTOMATION OPTIONS policy item for System Defaults (SDF). The default is 300.

```
------------------------------------------------------------------------------
                            System Automation Options

Entry Type : System Defaults       PolicyDB Name   : MYPDF
Entry Name : SYSTEM_DEFAULTS       Enterprise Name : MYPDF

System specific definitions:
  Exceptional Messages Limit . 300       (0 to 99999999, default 300)
  IPL Complete Time Limit. . . ____      (00:00:00 to 24:00:00)
  IPL Complete Status  . . . . ____________________________

                                         (AVAILABLE SOFTDOWN STANDBY)
                                         (or SATISFACTORY)
```

For details of the policy items mentioned in this topic, refer to the panel help by pressing PF1 or refer to the *Defining Automation Policy* manual.

Policy Definitions for Captured Messages

Messages of interest can be captured by making definitions in the policy using the Customization Dialog. If you only need to capture and display a message, but not to automate it, for APL and MVC you can list this message in the MESSAGES/USER DATA policy item and select **Capture** on the **AT Status Specification** panel to capture it.

Messages that have a CMD or REP action defined for them or that are defined as status messages are implicitly captured. There is no need to explicitly define these messages to be captured. In fact, doing so results in capturing them twice.

For example, to define message XYZ555I to be captured, enter AS option on the **Message Automation Overview** panel to display the **AT Status Specification** panel and select the **Capture** option.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                             AT Status Specification
Command ===>

View or change specifications for message: XYZ555I

Application / MVC Component status      Monitor health status
_ ACTIVE      Starting                  _ NORMAL   Resource shows good results
_ UP          Available                 _ WARNING  Resource shows degradation
_ HALTED      Degraded                  _ MINOR    More severe than WARNING
_ Terminating Received STOP command     _ CRITICAL More severe than MINOR
_ Terminated  Terminated normally       _ FATAL    More severe than CRITICAL
_ ABENDING    Going to end abnormally   _ UNKNOWN  Health status not available
_ Abended     Ended abnormally          _ Check    Evaluate health status
_ BREAKING    Non-recoverable abending  Monitor status
_ BROKEN      Non-recoverable abend       FAILED   Monitor failed
                                          BROKEN   Monitor finally failed
Capture specification
s Capture     Capture Message
              DOM . . . _________________________________
```

This definition leads to the creation of an AT entry for message XYZ555I using the AOFCPMSG command after the next configuration build process, as shown on the `Message Automation Overview` panel.

```
  COMMANDS  VIEW  HELP
------------------------------------------------------------------------------
                         Message Automation Overview

 AC AT condition     MS MRT action selection     MF MPFLSTxx specifications
 AS AT status        MO MRT override             IG Ignore for AT, MRT, MPF
 AO AT override

Message ID : XYZ555I

Generated AT entry :
IF MSGID = 'XYZ555I' THEN
 EXEC(CMD('AOFCPMSG') ROUTE(ONE %AOFOPGSSOPER%));

Generated MRT entry :
UPON ( MSGID = 'XYZ555I' )
REVISE('Y' AUTOMATE)

Generated MPF entry :
XYZ555I                                                                  as
```

At the same time, you can also delete (DOM, Delete Operator Message) an existing message by specifying its ID in the DOM field. In the example above, to delete message XYZ666I upon capturing XYZ555I, select the **Capture** option and specify XYZ666I in the **DOM** field.

```
Capture specification
S Capture     Capture Message
              DOM . . . XYZ666I_________
```

This definition leads to the creation of an AT entry for message XYZ555I using the AOFCPMSG command with parameter DOM=XYZ666I after the next configuration build process, as shown on the **Message Automation Overview** panel.

```
Generated AT entry :
IF MSGID = 'XYZ555I' THEN
 EXEC(CMD('AOFCPMSG DOM=XYZ666I') ROUTE(ONE %AOFOPGSSOPER%));
```

For more information about AOFCPMSG, see "AOFCPMSG" in *Programmer's Reference*.

Messages captured this way will always have a default severity of UNUSUAL. To overrule this default, you can use the so-called pseudo message CAPMSGS. See "Pseudo message ID CAPMSGS" on page 41.

Implicit Messages Capturing

SA z/OS captures certain messages implicitly without any user definitions in the following cases:

- A threshold (infrequent, frequent, or critical) is reached for a command, an APL or a minor resource.
- A command fails with a return code larger than 0 and return code checking is turned on.
- The APL status changes.
- WTORs are retained for reply processing.
- CICS transactions are recovered for short-on-storage conditions.
- IMS transaction or program abnormally end.
- DB2 recovery takes place.

In all cases, SA z/OS determines the severity based on the actual condition detected. If you do not want SA z/OS to capture any messages implicitly, you can create definitions within CAPMSGS pseudo message entry in the policy and assign the severity IGNORE to those messages. If you want a different severity from the one SA z/OS uses implicitly, you can also use CAPMSGS pseudo message. See "Pseudo message ID CAPMSGS" on page 41.

Explicit Messages Capturing

If you want to capture messages which are not implicitly captured by SA z/OS, for example in your own AT entry or within a user-defined script, you can use the AOFCPMSG command. See also "Policy Definitions for Captured Messages" on page 35.

Determine Where Captured Messages Are Sent

If you want to present captured messages not only on the NetView panels displayed by the operator commands, you can use the **Inform List** field in the policy to specify one or more targets that the message can be relayed to. You can specify this field for individual resources (APL, APG, MTR), MVS component defaults (MDF), system defaults (SDF), sysplex defaults (XDF), or in the CAPMSGS pseudo message in Message/User Data in policy items (APG, APL, MTR, MVC).

Inform list	Description
SDF	Exceptional messages are forwarded to the Status Display Facility.
EIF, IOM, USR, or TTT	All captured messages (if they have ALERT=YES specified when they are captured) can be transformed into particular alerts and forwarded to the corresponding target based on the INGALERT settings.

Deleting Captured Messages

While SA z/OS offers no means to delete captured messages from the resource-related buffers (for example, DISPINFO), it does support the management of exceptional messages.

Message entries in the exceptional message buffer are automatically cleared in the following cases:

- A message is deleted by Delete Operator Message (DOM).
- A SA z/OS-managed resource on whose behalf exceptional messages are held is shut down.
- A user deletes a message from the Status Display Facility.
- The INGMSGS command is used to delete individual or groups of exceptional messages.

In case an exceptional message is deleted, a clearing event is also automatically created via INGALERT, if a problem event has been sent.

Pseudo Messages

SA z/OS uses the concept of *pseudo messages* to implement certain automation solutions. These pseudo messages are defined in MESSAGES/USER DATA Policy Item for APLs, APGs, MTRs, and MVCs like any other 'real' message IDs.

The following pseudo messages are provided by SA z/OS:

- A special set of pseudo messages, which are used to invoke commands when an APL enters a certain status. See “Pseudo Messages to Invoke Status Commands” on page 38.
- Certain pseudo messages, which are used to invoke commands for a consumer APL when its provider APL reaches an "up" or "down status. See “Pseudo Messages to Invoke Extended Status Commands” on page 39.
- Pseudo message ID CAPMSGS, which allows you to overrule the specified or default values of AOFCPMSG. See “Pseudo message ID CAPMSGS” on page 41.
- Pseudo message ID INGALERT, which allows you to define additional characteristics for events to be passed to the notification target or to prevent event creation for certain alerts. See “Pseudo message ID INGALERT” on page 42.
- Other pseudo message IDs for diverse automation solutions. See “Other Pseudo Message IDs” on page 42.

Pseudo Messages to Invoke Status Commands

A special set of pseudo messages are used to invoke commands at the time an APL enters a defined status, for example UP status. These commands are called status commands.

The following pseudo message IDs are valid:

- ABENDING
- ACTIVE
- AUTODOWN
- AUTOTERM
- BREAKING
- BROKEN
- CTLDOWN
- DOWN
- ENDED
- ENDING
- EXTSTART
- FALLBACK
- HALFDOWN
- HALTED
- MOVED
- RESTART
- RUNNING
- STARTED
- STARTED2
- STOPPED
- STOPPING
- STUCK
- UP

- ZOMBIE

These pseudo messages correspond to the automation agent statuses. For more information, see “Application Statuses Supplied by Automation Agents” on page 70.

Pseudo Messages to Invoke Extended Status Commands

The status command concept has been extended so that commands can be issued for an APL if another APL reaches an "up" or "down" status.

An "up" status in this context is an automation agent status of:

- UP
- ENDED

A "down" status in this context is an automation agent status of:

- DOWN
- RESTART
- AUTODOWN
- STOPPED
- CTLDOWN
- BROKEN

Thus, a command can be issued for application APL1 when application APL2 enters or is already in a certain status. Application APL1 is called the *consumer* APL that consumes services provided by application APL2, which is called the *provider* APL. In certain cases, it is valuable to trigger a consumer action if the provider enters an "up" or "down" status. Alternatively, when the consumer enters an "up" status, it can take an action depending on the status of the provider.

In addition to the consumers runtime variables, there are provider-specific runtime variables that can be used within a command or reply as specified for a Message ID in the MESSAGES/USER DATA policy item. These variables start with &SU2 instead of &SUB. For a list of supported consumer and provider runtime variables, see ACFCMD and ACFREP in *Programmer's Reference*.

Extended status commands are defined in the MESSAGES/USER DATA policy item for the *consumer* APL. The extended status command denotes the subsystem name of the *provider* APL and the status of interest:

- The consumer name is the Subsystem Name of the *consumer* APL.
- The provider name is defined in the MESSAGES/USER DATA policy item of the *consumer* APL as part of a pseudo Message ID.
- To execute a command by a *consumer* APL whenever a *provider* APL enters an "up" or "down" status,
  - Use UP_*provider-subsys* to define the commands that are executed when *provider-subsys* enters an "up" status.
  - Use DN_*provider-subsys* to define the commands that are executed when *provider-subsys* enters a "down" status.

Other pseudo messages are available to hold sets of commands on a *consumer* APL that are executed whenever the *consumer* APL enters an "up" status and the *provider* APL is already in an "up" or "down" status:

- Use ISUP_*provider-subsys* to define the commands that are executed when the *provider-subsys* is in an "up" status.
- Use ISDN_*provider-subsys* to define the commands that are executed when the *provider-subsys* is in a "down" status.

No commands are issued when the *consumer* APL enters a "down" status.

You can create a so-called dynamic link by defining a USER action with the Keyword/Data pair DYNAMIC=YES for the pseudo message ID. For dynamic links, the commands are executed when the link is activated via INGLINK, and the provider is in the appropriate status. For more information, see INGLINK in *Programmer's Reference*.

For example, if you want to start the MQ Listener for the MQCHIN application whenever the TCPIP application reaches an "up" status, you define the extended status command for the message ID UP_TCPIP using the MESSAGES/USER DATA policy item of MQCHIN. On the **Message Processing** panel, enter the line command "Cmd" for the UP_TCPIP message ID and on the subsequent **Command Processing** panel, enter `MVS START LISTENER` in the **Command Text** field.

```
                          Message Processing       Line 00000001 Col 001 075

Entry Type : Application          PolicyDB Name   : PDBDEMO
Entry Name : MQCHIN               Enterprise Name : PDBDEMO

Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)
               I, D (insert or delete lines), CN (Class Name)
Message ID field length. . 16   (1 - 32)

Cmd Message ID       Description                              C  R  K  U A M F
cmd UP_TCPIP         Start sockets when TCPIP is UP
```

```
            Command Processing : UP_TCPIP            Line 00000001 Col 001 075

Mixed case . . . NO   (YES NO)

Cmd Ps/Select  AutoFn/* Command Text
___ _________  ________ MVS START LISTENER
******************************** Bottom of data ********************************
```

For dynamic links, you need to define pseudo message IDs for all the providers that might be used by a consumer. There are several alternatives for activating a dynamic link:

- In the POSTSTART phase of the STARTUP policy item
- Based on any message before the consumer is in the "up" status
- Based on the "up" status
- Based on any message after the consumer is in the "up" status

You can make these definitions in either the STARTUP or MESSAGES/USER DATA policy item. Corresponding policy definitions are required for link deactivation either in the SHUTDOWN or MESSAGES/USER DATA policy item.

You must supply a user-written REXX automation procedure that:

1. Identifies the provider.
2. Issues the INGLINK command to activate the link, if required.

All the dynamic links of a consumer are automatically deactivated after the consumer is shut down. Thus, you only need to define INGLINK DEACTIVATE commands if you want this process to happen at an earlier point in time.

There is no difference in the DN_ and UP_ message processing between normal and transient applications, with one exception: when stopping a transient provider, the corresponding DN_ message on the consumer is *not* processed.

Special Considerations

You can define a consumer as its own provider. This can be done to perform a predefined action for an application whenever it enters a "down" status.

Furthermore, you must not define an INGLINK DEACTIVATE if the consumer is its own provider, since this INGLINK DEACTIVATE may take place before the DN_ message can be processed.

Thus, you no longer need to define the same action for each of the agent statuses. To achieve this, define the same name in the DN_*provider-subsys* message and subsystem name field of an application policy entry.

Note:

- If the consumer and provider applications are different, the action defined under DN_*provider-subsys* is executed only when the consumer application is in an "up" status. However, if the consumer and provider applications are the same, the action defined under the DN_*provider-subsys* is executed even when the consumer application is not in an "up" status.
- Dynamic link definitions are not required. If you define a dynamic link, it is ignored.
- The action defined for the DN_*provider-subsys* message is not executed if the status change resulted from a failed startup.

Pseudo message ID CAPMSGS

With pseudo message CAPMSGS, you can overrule the specified or default values of AOFCPMSG.

In the code match table for pseudo message CAPMSGS, you list all the messages for that APL or MVC with the following attributes:

- Message ID
- Jobname
- User (CODE3)
- Severity
- Alert
- Inform List
- Comment type
- Comment Text
- DOM list

```
  COMMANDS  HELP
------------------------------------------------------------------------------
            Code Processing : CAPMSGS               Line 00000010 Col 001 075

Use line command E or S to display complete input fields of a single data line.

Cmd Message ID      Jobname  User (CODE3)    Severity  Alrt Inform List
___ _____________ _______ _____________ _________ ____ ___________
****************************** Bottom of data ******************************
```

Code processing works from top to bottom and ends at the first row that matches. If there is a match with Message ID, Jobname, and User, the specified values of the other fields are used to set up the parameter values for AOFCPMSG.

If a message has been captured and the "Alert" specification in the CAPMSGS code match table is found to be YES, AOFCPMSG triggers INGALERT.

If no matching rule is found, the ALERT parameter specified in the AOFCPMSG call is used. If the ALERT parameter value is also missing, AOFCPMSG invokes INGALERT for each message with a specified severity of CRITICAL.

For more information about AOFCPMSG and INGALERT, see *Programmer's Reference*.

Refer to the MESSAGES/USER DATA Policy Item in *Defining Automation Policy* for more details about defining message entries, capturing them with the AT Status option and how to use code processing in the Customization Dialog.

Pseudo message ID INGALERT

With pseudo message INGALERT, you can define additional characteristics for events that are passed to the notification target or to prevent event creation for certain alerts.

You need to add pseudo message INGALERT to the list of messages and use the code processing.

Make the following specifications in the code match table for pseudo message INGALERT:

- "Code 1" for the alert ID.
- "Code 2" for the job name.
- "Code 3" for the communication method used to send the alert to the notification target.
- "Value Returned" for parameters that are sent to the notification target.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
            Code Processing : INGALERT               Line 00000001 Col 001 075

Use line command E or S to display complete input fields of a single data line.

Cmd Code 1          Code 2          Code 3          Value Returned
___ _______________ _______________ _______________ ____________________________
___ _______________ _______________ _______________ ____________________________
******************************* Bottom of data ********************************
```

Code processing works from top to bottom and ends at the first row that matches. If there is a match with **Code 1**, **Code 2** and **Code 3**, the specified values of the **Value Returned** field is used as data for the notification target.

For more information about alerting, see chapter "Alert-Based Notification" in *Customizing and Programming*.

Other Pseudo Message IDs

To implement other automation solutions, use the following pseudo message IDs:

Table 5. Pseudo message IDs

Pseudo message IDs	Usage
ABCODEPROG	IMS program abend codes handling
ABCODETRAN	IMS/CICS transaction abend codes handling
AMRFCLEAR	AMRF recovery
AMRFFULL	AMRF recovery
AMRFSHORT	AMRF recovery
CAPMSGS	See “Pseudo message ID CAPMSGS” on page 41.
CICSINFO	CICS display information
IMSINFO	IMS display information
INGALERT	See “Pseudo message ID INGALERT” on page 42.
LISTSHUT	CICS transaction purging
LOGREC	LOGREC processing commands
MVSDUMP	MVSDUMP threshold commands
MVSDUMPRESET	MVSDUMP reset commands
MVSDUMPTAKEN	MVSDUMPTAKEN commands

Table 5. Pseudo message IDs (continued)	
Pseudo message IDs	**Usage**
OLDS	IMS OLDS monitoring and recovery
OPCA	ZWS request definitions (status and interval)
OPCACMD	ZWS request definitions (commands)
OPCAPARM	ZWS request definitions (modifications)
RCVRSOS	CICS short on storage recovery
RCVRTRAN	CICS Transaction recovery
SHUTFORCEDDF	DB2 stop threads
SMFDUMP	SMFDUMP commands
SPOOLFULL	JES2 or JES3 Spool recovery
SPOOLSHORT	JES2 or JES3 SPOOLSHORT commands
SYSLOG	SYSLOG processing commands
SYSTEM_SHUTDOWN	System shutdown processing
TCO	IMS time-driven procedures
TCOMEMBERS	IMS Timer Controlled Operations
VTAMUP	INGVTAM definitions
WORKSTATION	ZWS command I/F workstation definitions
WTOBUF	WTO buffer shortage automation
WTORS	WTORs classification

Defining AT Conditions and Actions, MRT and MPF

You can define various actions for messages using the **Message Automation Overview** panel.

- Status changes for messages (see “Defining Messages for Status Change (Status Messages)” on page 32).
- Messages to be captured (see “Defining Messages for Capturing” on page 33).
- AT condition and override (see “Defining AT Conditions” on page 43 and “Defining AT Overrides” on page 44).
- MRT condition and override (see “Defining MRT Conditions and Overrides” on page 44).
- Preventing the building of AT, MRT and MPF entries (see “Inhibiting AT, MRT, and MPFLSTxx Entries” on page 45).
- MPF specifications.

Defining AT Conditions

You can improve the efficiency of AT processing by controlling where entries are placed within the AT and by specifying more precise conditions to trap the message.

SA z/OS allows you to do so with the **Automation Table entry Conditions** panel, which you reach from the **Message Automation Overview** panel by entering the option AC.

```
  COMMANDS  HELP
-------------------------------------------------------------------------------
                        Automation Table entry Conditions

 View or change specifications for message: ABC134I

 Ignore for AT . . . . . . . . . ____     (YES NO)
 Message placement . . . . . . . _______  (TOP BOTTOM)
 Check for message ID  . . . . . ____     (YES NO)
 Ignore message ID characters. . ________ (LEADING TRAILING BOTH)
 Additional condition  . . . . .
 ______________________________________________________________________________
______________________________________________________________________________
```

Defining AT Overrides

You can apply an AT override for a message ID for an APL instance, an APL class, a MTR, or an MVC entry. The option "AO" on the **Message Automation Overview** panel allows you to override an AT entry. You can change any part of the AT entry, either changing, adding, or deleting the condition or action statements.

If you define a message with an AT action or condition, and then invoke the override panel, the preview of the AT or MRT entry is shown on the override editor screen. You can use this preview as a model for your own AT or MRT definition. To exit the editor without saving your changes, use the CANCEL command.

If you specify an AT status selection for a message with an AT override, then a confirmation panel for the "override delete" is displayed because an override cannot be combined with the other specifications.

```
            Confirm AT Override Delete

 An AT override is already specified for this message.
 However the new specification on this panel conflicts
 with the existing override which will be deleted.

 Press ENTER to delete AT override.
 Press CANCEL or END to keep AT override.

  F1=Help    F2=Split   F3=End     F9=Swap   F12=Cancel
```

You can include system automation symbols (AOCCLONE variables) and system symbols in an AT override definition. They are resolved at AT load time.

You can define '&*JOBNAME.' as part of an AT condition statement that will be replaced by the job name of the given policy entry when building the AT. This is valuable when defining an AT entry for an APL class. Then each APL instance linked to that class will have its own AT entry with its job name in the AT condition statement. Checking for the job name may also be required if different instances of a product issue the same message but you want only certain jobs to be affected by that message.

It is possible to enable syntax checking for AT and MRT. For more information, see "Enabling and Disabling AT / MRT Syntax Checking" in *Defining Automation Policy*.

Defining MRT Conditions and Overrides

The MRT enables user-defined modification of attributes such as color, route code, descriptor code, display and syslog settings, and text of original z/OS messages.

You can make decisions about the message based on its message ID, job name, and many other properties. You can have only one MRT active per system.

Any MRT specification that you make are independent of any AT entry data for the message. Thus, if you make a definition for the MRT, the existing AT data remains unchanged.

The **MS** option on the **Message Automation Overview** panel allows you to define conditions and attributes that are used to generate MRT entries for a message ID. Use the fields on the offered **Message Revision Table Conditions** panel to specify the following conditions:

- Ignore the message for the generated MRT.

- Delete the message completely (if you select this condition, no other selection is valid).
- Automate the message.
- Suppress the message from the console or system log.
- Translate the message text to uppercase or append further text to it.
- Change the color, highlighting, and intensity (if your terminal supports high intensity) attributes of the message. Only one selection for each of these attributes is allowed.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFGMRT1               Message Revision Table Conditions
Command ===> ____________________________________________________________

View or change specifications for message: IKT007I

Ignore for MRT . . . . . ___          (YES NO)
Delete completely  . . . . ___        (YES NO)

Automate . . . . . ___          (YES NO)
Suppress from console  . . ___        (YES NO)
Suppress from system log . ___        (YES NO)
Translate to upper case  . ___        (YES NO)
Append text  . . . . . ____________________________________________________


Color  . . . . . _______      (BLUE RED PINK YELLOW GREEN WHITE TURQ)
Highlighting . . . . . _____________ (UNDERLINED BLINKING REVERSEVIDEO)
Intensity  . . . . . ___      (YES NO)
```

You can include system automation symbols (AOCCLONE variables) and system symbols in an MRT override definition. They are resolved at MRT load time.

A syntax check is not made of the MRT entry because any system-specific definitions (for example, symbols) can be verified only on the system where the MRT is to run.

You should check that the routines triggered by the AT entry for the message ID are compatible with any changes you made to the original message text.

You can also use the **MO** option on the **Message Automation Overview** panel to define the MRT entry directly using a full-screen editor. A syntax check is not performed on this panel. Ensure that your specifications follow NetView message revision table syntax rules.

For more information, see The Message Revision Table in *IBM Z NetView Automation Guide*.

Inhibiting AT, MRT, and MPFLSTxx Entries

Messages that are marked IGNORE do not cause an AT entry, MRT entry, or an MPFLSTxx entry to be generated. To ignore a message, enter **IG** option on the **Message Automation Overview** panel to display the **AT/MRT/MPF Ignore Specification** panel, then select the **Ignore** field.

```
  COMMANDS  HELP
------------------------------------------------------------
                        AT/MRT/MPF Ignore Specification
Command ===>

View or change specifications for message: NEWMESG

  S Ignore      No AT entry; no MRT entry; no MPFLSTxx entry
```

IGNORE discards other actions that are defined for the same message.

You can also specify a selective IGNORE for AT, MRT, and MPF only. For example, you can specify to ignore the message for MPF only but not for AT and MRT. To do so, enter MF command on the **Message Automation Overview** panel to display the **MPF Specification** panel, then specify yes in the **Ignore for MPF** field.

```
                                MPF Specification
Command
===>_____________________________________________________

View or change specifications for message: NEWMESG

MPF default specification from policy MPF DEFINITIONS:
  .DEFAULT,SUP(YES),RETAIN(I,CE),AUTO(YES)

Ignore for MPF . . . . . . YES        (YES NO)

Alternate message ID . . . __________
Message parameters . . . . ________________________________________
```

AT/MRT/MPF Build and Activate

This section describes how to build and activate AT, MRT, and MPF members, and how to view generated AT and MRT listings.

Building AT/MRT and MPF

After you make all the message definitions that you need, you can start the configuration build process to build the AT, MRT, and MPF members.

Build function is available in option 2 (Build) on the Customization Dialog primary menu. For more information about the build function, see chapter "Building and Distributing Configuration Files" in *Defining Automation Policy*.

The AT, MRT, and the MPFLSTxx members are built into the configuration build process output data set. They are automatically included in a "Build" with type "MODIFIED" if changes have been made which affect those members.

Note: If the MPF Header or Footer definitions have changed, run a build with type "ALL" to make sure that these changes are incorporated by building MPFLSTxx anew.

Activating AT/MRT

After the ATs and MRT members are generated using the Customization Dialog, they are ready to be loaded.

Configuration refresh can be used to refresh the complete SA z/OS configuration and the related ATs and MRT as they are defined in the SA z/OS policy database. Alternatively, you can load ATs by using the ACF ATLOAD command or the NetView AUTOTBL command.

The advanced automation option (AAO) AOFSMARTMAT controls whether the AT and MRT members generated at configuration build process should be used. For compatibility reasons, the provided default is 2 indicating that the generated AT member (but not the MRT member) is loaded at SA z/OS initialization time or during configuration refresh. The recommended value of AOFSMARTMAT is 3 indicating that both the generated AT and MRT members are loaded at SA z/OS initialization time or during configuration refresh. For more information about AOFSMARTMAT, refer to "Read/Write Variables" in *Customizing and Programming*.

Some AT entries are required for SA z/OS to operate properly. These entries reside in a separate AT that is loaded during SA z/OS initialization. This AT is called INGMSGSA. Do not edit it.

Generated AT/MRT Listings

Whenever SA z/OS loads an AT or MRT, a listing is generated and stored in the DSILIST data set.

If you want to view the listing of the AT INGMSG01, issue the command:

```
br dsilist.ingmsg01
```

To view the listing of the MRT, issue the command:

```
br dsilist.ingmrt01
```

If you want to suppress listings, you can set the advanced automation option (AAO) AOFMATLISTING to zero. For more information about AOFMATLISTING, see "Read/Write Variables" in *Customizing and Programming*. Without these listings, you can monitor or manage the ATs only by the NetView AUTOTBL command.

Having listings available enables you to use both the NetView AUTOMAN and AUTOTBL commands to monitor and manage the ATs. Make sure that the size of your DSILIST data set is sufficient to store these listings and each NetView uses its unique DSILIST data set. It is recommended that you define your DSILIST data set as a PDSE.

If the testing or loading of an AT or MRT fails, a special INGERRLS listing that contains the data of the failing AT or MRT is written to DSILIST. To view this listing, issue the command:

```
br dsilist.ingerrls
```

Concept of SA z/OS Automation Tables

This section describes the structure of the ATs supplied by SA z/OS , how to integrate user-written AT, and how to view the status of the activated ATs.

AT Structure

All entries for the SA z/OS basic automation infrastructure are provided via AT INGMSGSA. This AT is always loaded during SA z/OS initialization and must not be edited.

Additionally, SA z/OS provides a ready-to-use AT, named INGMSG01, which is suitable for use as a primary AT. The INGMSG01 AT contains:

- AT entries for messages that are defined in the PDB
- User include members

To activate INGMSG01, follow these steps:

1. Define the AT member INGMSG01 in the SYSTEM INFO policy of every system in the Customization Dialogs. It is the default value if nothing is specified. If you want to specify any additional user-defined AT, make sure that INGMSG01 is in the list of names. If not, INGMSG01 will NOT be loaded, which can cause major problems.
2. Run a configuration build.
3. Refresh the configuration using INGAMS REFRESH or restart NetView with the new configuration.

Do not edit the AT INGMSG01. All unused entries are disabled automatically according to the configuration that you use. If you want to have additional entries that are valid only for your environment, you can use either a separate AT (specified in SYSTEM INFO policy item) or use one of the user-defined include members.

Figure 12 on page 47 shows the structure of the AT:

```
INGMSG01
├── %INCLUDE AOFMSGSY (SA supplied synonyms)
├── %INCLUDE INGMSGU1 (user include member)
├── %INCLUDE INGMSG02 (generated by configuration build process)
└── %INCLUDE INGMSGU2 (user include member)
```

Figure 12. AT Structure

The following fragments are used by the AT:

Synonym Definitions
There is one member, AOFMSGSY, that is used to initialize the various synonyms used throughout the rest of the table. SA z/OS requires the synonyms to be suitably customized to reflect your environment. For more information about the synonyms, see "Generic Synonyms: AOFMSGSY" in *Customizing and Programming*.

SA z/OS Functional Definitions
These definitions, which are located in the INGMSG02 member, contain AT statements depending on the definitions in the policy. You can use the Customization Dialog to override the SA z/OS predefined AT definitions or use the separate ATs INGMSGU1 and INGMSGU2 for any special modifications.

INGMSG01 is the main include member. It provides some message suppression that is necessary to prevent mismatches and duplicate automation before the first %INCLUDE member.

The INGMSGU1 member can be used for user entries. These entries have precedence over the SA z/OS entries. The default INGMSGU1 member is empty.

The INGMSGU2 member can be used for all entries that SA z/OS does not provide any entries for. The default INGMSGU2 member is empty.

During configuration refresh, ATs are loaded and a listing is written to the DSILIST data set. See “Generated AT/MRT Listings” on page 46.

Integrating User-written ATs

SA allows you to include user-written ATs. The recommended approach is to use the SYSTEM INFO policy item to add them to the list of ATs automatically loaded by SA z/OS.

Alternatively, you can use the user-defined include members INGMSGU1 and INGMSGU2.

Finally, user-written ATs can be maintained also outside of SA z/OS by using the NetView command AUTOTBL. For more information about this command, see AUTOTBL in *IBM Z NetView Command Reference Volume 1 (A-N)*.

Displaying Status of Activated ATs

To display the status of the activated ATs, use the NetView **AUTOTBL STATUS** command.

```
BNH361I THE AUTOMATION TABLE CONSISTS OF THE FOLLOWING LIST OF MEMBERS:
IPXFIPPT COMPLETED INSERT  FOR TABLE #1: DSITBL01 AT 11/08/18 09:19:43 (FIRST)
DSI#0976 COMPLETED INSERT  FOR TABLE #3: INGMSGSA AT 11/14/18 11:22:35
DSI#0978 COMPLETED INSERT  FOR TABLE #4: INGMSG01 AT 11/14/18 11:22:35

IPXFI
BNH363I THE AUTOMATION TABLE CONTAINS THE FOLLOWING DISABLED STATEMENTS:
TABLE: INGMSGSA INCLUDE: __n/a___ GROUP   : INGJES3
-- - - - - - - - - - - - - - ---- - - - - - - - - - - - - - - - - --
TABLE: INGMSG01 INCLUDE: __n/a___ GROUP   : INGCICS
TABLE: INGMSG01 INCLUDE: __n/a___ GROUP   : INGJES3
```

Or you can use the **AUTOMAN GROUP** command. Here is an example of the AUTOMAN panel:

```
EZLKATGB              AUTOMATION TABLE MANAGEMENT

MEMBER    TYPE   LABEL/BLOCK/GROUP NAME(S)  STATUS    NUMBER OF STATEMENTS
--------  -----  -------------------------  --------  --------------------
INGMSG02  GROUP  INGCICS                    DISABLED          13
INGMSG02  GROUP  INGIMS                     ENABLESD           3
INGMSG02  GROUP  INGJES2                    ENABLED            1
INGMSG02  GROUP  INGJES3                    DISABLED           1
```

This example shows that the groups INGCICS and INGJES3 are disabled in AT INGMSG02 and hence the configuration does not use the CICS and the JES3 product automation, respectively. It uses IMS and JES2 automation.

Generic AT Statements

The ATs supplied by SA z/OS contain a number of generic AT entries that can reduce your AT overhead considerably. These samples use some of the advanced features of SA z/OS to make automating your applications as simple and reliable as possible.

For some of these entries (IEF403I and IEF404I in particular), the message flow can be quite high. To handle it, you can insert additional entries in INGMSGU1 to suppress a block of messages. For example, if all your batch jobs started with the characters BAT or JCL, then the following entry would suppress them:

```
IF MSGID = 'IEF40'. & DOMAINID = %AOFDOM% THEN BEGIN;
*
  IF (TOKEN(2) = 'BAT'. | TOKEN(2) = 'JCL'.)
    THEN DISPLAY(N) NETLOG(N);
*
END;
```

End-to-End Automation on z/OS

SA z/OS automation manager is able to automate resources between multiple sysplexes (or SAplexes) and cross platforms. This cross sysplex/platform automation is called end-to-end automation. In this case, the primary automation manager (PAM) is also called E2E manager.

In conjunction with the E2E manager, there is a new address space, which represents the E2E agent. The E2E agent assists the E2E manager by executing start and stop orders or collecting status of resources running on remote automation domains. For that purpose, the E2E agent interfaces with the remote automation adapter.

Via the Customization Dialog, you may define new objects with type DMN and REF. The DMN object represents a remote domain and the REF object represents a remote resource.

With end-to-end automation, you can

- Start and stop remote resources.
- Monitor remote resources.
- Manage cross domain dependencies between local and remote resources or between two remote resources.
- Compose business applications that have high availability across multiple automation domains.
- Manage a group of remote resources.

You must activate the end-to-end automation first before using it. To activate and use the end-to-end automation, you must meet the following basics:

1. Defined REF and DMN objects in the currently active policy loaded by the PAM.
2. Started a new address space, which represents the E2E agent.
3. Started an end-to-end automation adapter on each remote domain for which a DMN object exists.

For more information, see

- *IBM Z System Automation Defining Automation Policy*
- *IBM Z System Automation End-to-End Automation*
- *IBM Z Automation Web Console Installation and Configuration Guide*

Figure 13 on page 50 shows that end-to-end automation adapter for z/OS connects multiple remote SA z/OS automation domains to the E2E agent. The E2E agent enables the PAM to takeover the role of the end-to-end automation manager.

This figure also shows that SA z/OS can be connected to an Z Automation Web Console server and the Universal Automation Adapters can be connected to the Z Automation Web Console server. The web-based dashboard is used to view all automation domains and resources.

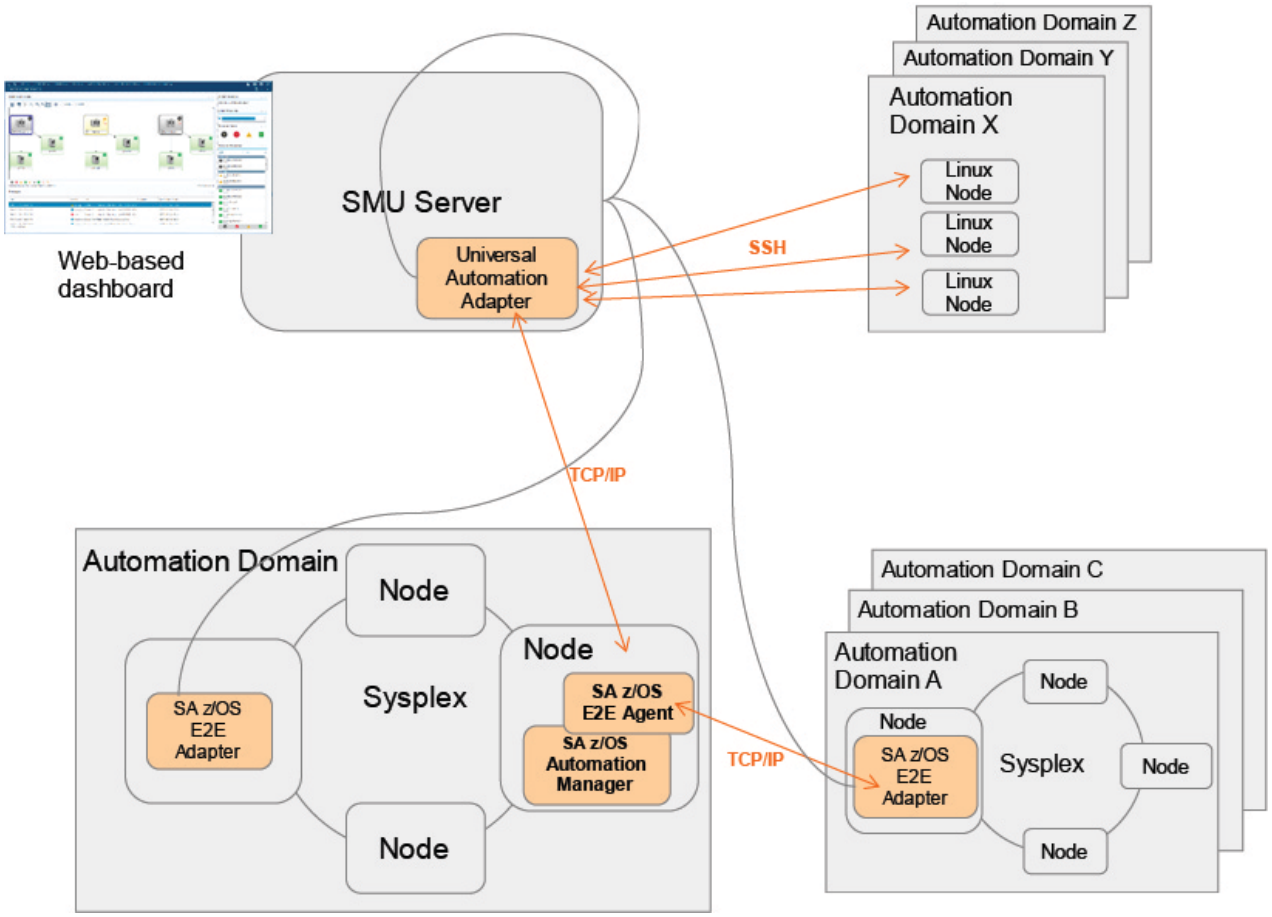


Figure 13. End-to-End Automation on z/OS

Groups

Groups are active automation elements that manage the availability of a small set of resources.

For example, a group might be all the resources needed to run a CICS or an IMS or, at a higher level, a complex application, relying on subgroups to manage the complex resources that are needed to make it work. The advantages of groups are that they automatically manage the availability of the resources they are looking after and provide you with a single point of control to request that resources are stopped or started. The down side is that if an operator does not tell them what they want them to do, they can end up working at cross purposes.

Group Types

There are four types of groups available within System Automation - Basic, Move, Server and Simple.

Basic Groups

These groups manage the resources defined to them as if they were a single resource. Ask the group to start and it starts them all, ask it to stop and it stops them all. If one of them fails and cannot be restarted, it concludes that the group as a whole is broken, which can lead to it being shutdown. This can also happen if the manager automation flag for one or more group member is turned off.

Move Groups

These groups manage the availability of a single resource, choosing between alternative instances of it on different systems. The resource may be a single APL or may be a group of resources. The group is responsible for picking the system to start it on and for 'moving' it to other systems as required to maintain its availability. A Move group will not pick a member that cannot be started. If its members are basic groups, this means that if a single member of the basic group has a problem, the Move group will not pick the entire Basic group.

Server Groups
Like Move Groups, Server Groups manage the availability of resources over multiple systems. A Server Group gets a set of resource instances and a target number of them to keep available. It is responsible for picking the instances that originally get started and for 'moving' them around to maintain availability. Like Move Groups, Server Groups will not pick members that cannot be started.

Simple Groups
These groups are different, in that they are a construct only used in the Customization Dialogs. They do not appear as a resource. They are differentiated from other group types by the absence of an automation name. Simple groups are used to simply place resources on systems and provide no ongoing management of them once automation is running. Resources with Simple Groups are managed mostly by their Default Desired Status, propagated votes and schedules.

Relationships

Relationships upon a group work on the group, and are usually passed onto the group members as well. This means that if a group is a child of a resource, then every member of the group is also considered to be a child of the resource, at least from the point of view of working out what can be stopped and started.

Note that there is an exception to this for Move and Server groups which allow their members to be stopped (provided the group as a whole is not being stopped) while their Stop dependency is unsatisfied. This is necessary to provide their ability to change their set of active members.

Runmodes

Runmodes are not groups, but they can provide a similar function at some levels. Runmodes let you 'tag' a resource with a runtoken, and then define runmodes as sets of runtokens. The INGRUN command can be used to activate or deactivate particular runmodes. Resources whose runtokens are not selected get shutdown.

It is strongly suggested that runmodes rather than large groups containing unrelated resources be used for 'bulk' application control on a system.

To learn more about runmode and try this feature in a simulated System Automation environment, check this hands-on tutorial.

Operating Groups

You should start and stop groups with the same commands, INGREQ and INGVOTE, as you use to operate normal APL resources. These simply request that the group be started or stopped according to its predefined priorities.

In the same way you can use INGSUSPD and INGVOTE to suspend and resume automation for groups as well as for APLs.

Group Recovery Mode

IBM Z System Automation can automate recovery of z/OS resources, components, and applications to minimize the impact on outage. This function applies to Move and Server groups only because all members should be started for Basic groups. Move and Server groups can enter and leave the recovery mode.

Recovery mode refers to a state that a set of actions are initiated to restore normal operation and recover from system or application failures or other critical issues in the mainframe environment. For example, if a primary member becomes unavailable, a backup member is then started to maintain continuity of services and minimize downtime. System Automation 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.

A group enters the recovery mode in any of the following conditions:

- When its previously selected and available member unexpectedly goes into the status (Stopping, Problem, HardDown, SoftDown, or SysGone).
- When its member with preference 600 or greater AND observed status (Available, Degraded, Starting, Stopping, WasAvailable, SoftDown) becomes (HardDown, Problem) or (SysGone) for a selected member.
- When the member's system gets excluded.

A group remains in the recovery mode in any of the following conditions:

- The number of selected members is less than the availability target.
- Any selected member has preference 599 or less.
- Any selected member in (SysGone, HardDown) observed status has preference 2000 up to 2599.

Moving MOVE Groups

You can use the INGMOVE command to issue Move directives. These completely override the group's default policy and force it to move to a specific system. Their effect persists until a specific command is issued to release the resource.

Suspending Groups

You should suspend groups with the same commands, INGSUSPD and INGVOTE, as you use to suspend normal APL resources.

Controlling Application Move and Server Management

Application move refers to the process of stopping an active set of resources and starting an alternative set in a coordinated fashion.

Server management refers to the starting and stopping of equivalent resources to maintain a desired level of server capacity. Although it will work with any type of resource, it is designed to work with servers that are loosely coupled to the applications using them. Loosely coupled in this sense means that applications using the servers will not be adversely impacted if one instance of the server group is stopped and another one is started.

SA z/OS uses the mechanism of grouping resources as described in *IBM Z System Automation Defining Automation Policy* to implement the move mechanism. That is, the groups of nature MOVE and SERVER obey all the move instructions that are described in this section, along with an availability target instruction that specifies how many members of a group should be available, if all of them are not required.

Group Recovery Mode

IBM Z System Automation can automate recovery of z/OS resources, components, and applications to minimize the impact on outage. This function applies to Move and Server groups only because all members should be started for Basic groups. Move and Server groups can enter and leave the recovery mode.

Recovery mode refers to a state that a set of actions are initiated to restore normal operation and recover from system or application failures or other critical issues in the mainframe environment. For example, if a primary member becomes unavailable, a backup member is then started to maintain continuity of services and minimize downtime. System Automation 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.

A group enters the recovery mode in any of the following conditions:

- When its previously selected and available member unexpectedly goes into the status (Stopping, Problem, HardDown, SoftDown, or SysGone).

- When its member with preference 600 or greater AND observed status (Available, Degraded, Starting, Stopping, WasAvailable, SoftDown) becomes (HardDown, Problem) or (SysGone) for a selected member.
- When the member's system gets excluded.

A group remains in the recovery mode in any of the following conditions:

- The number of selected members is less than the availability target.
- Any selected member has preference 599 or less.
- Any selected member in (SysGone, HardDown) observed status has preference 2000 up to 2599.

Move Functions

The move functions that you can specify against a group are:

- Whether you want new members to be started at the same time as old members are being stopped (*parallel move mode*), or the old member should be stopped completely before the new member is started (*serial move mode*).
- Whether the downtime during the move or switch of a resource should be kept as short as possible (*prepare move*).
- Automatic group management to maintain application availability. This includes the activation of backup/alternate resources in response to resource or system failure.
- Moving resources away from a system en masse, at either a group or subgroup level or a system level.
- Querying and dynamically changing a group's policy to stop one instance of a resource and start another.
- Estimating the impact of those changes before committing them.
- Performing actions resulting from the above changes in a delayed, non-disruptive fashion.
- Returning resources to their normal systems in a delayed, non-disruptive fashion.

You can initiate move functions using either the INGMOVE or INGGROUP command.

Using the INGMOVE Command

The INGMOVE command allows you to specify where a sysplex application group should be moved to. INGMOVE provides details of the observed status of sysplex application groups and those systems that at least one member of the sysplex application group would be able to run on.

You can use the INGMOVE command dialog to:

- Initiate a move to a specified system.
- Prepare the move of a sysplex application group.
- Reset the preference values to those defined in the policy database.
- *Box* the group. The currently active member receives a preference of 3200 so that if the system where it runs is terminated, no other group member is activated instead and the group becomes unavailable.

  This can be useful if, when IPLing the system that hosts the primary group member, moving that member to another system and back to its home system (that is, the system with the highest base preference value) later on takes longer than the entire IPL process.

Using the INGGROUP Command

The INGGROUP command offers the following methods to apply move functions to a group:

EXCLUDE/AVOID/INCLUDE
This method allows you to:

- Remove movable members from a system (exclude)
- Prohibit SA z/OS from activating movable members on a system (avoid)
- Undo either of the other two actions (include)

Availability Target (AVTGT)
This specifies the availability target that is to be set for the group. The value is the number of the group members that should be made available when the group has to be available. You can dynamically overwrite a group's availability target that is statically defined in the automation policy.

Preference (PREF)
This parameter specifies preference values for being available for the members of a group. You can dynamically overwrite the preference values that are statically defined in the automation policy for each member. More information about the preference value is provided in “Controlling Application Move with Preference Values” on page 54.

Satisfactory Target (SATTGT)
This parameter specifies the number of members in the group that must be active before the automation manager treats the group as being in a satisfactory state. The value must be a positive, decimal number. The parameter will be ignored for groups other than server groups. You can dynamically overwrite a group's satisfactory target that is statically defined in the automation policy.

RECYCLE/CANCEL
This causes a *rolling recycle* to be initiated for each selected server or move group. This consists of sequentially stopping and then restarting each member of the group, so that at the end of the recycle all members that were active when the operation was requested have been stopped and recycled or stopped and left down. The CHUNK parameter specifies the number of server group members to recycle in parallel. Specifying a chunk size so that only a few passes are necessary to recycle the entire server group will significantly improve the performance.

Controlling Application Move with Preference Values

Controlling application move and server management is done with preference values. Preference values range from 0 through 3200. They express the importance of each individual member of the group.

Note that only MOVE groups and SERVER groups support preference values. With BASIC groups you cannot specify preferences, because all members need to be available. For MOVE and SERVER groups, the preference defines which resource members of the group should preferably run. Remember that a MOVE group is designed for backup solutions. A MOVE group can have many components, but one and only one is supposed to be active. All other components are for backup only. Thus MOVE groups usually contain members of one kind (where one application can replace the others).

SERVER groups are created when only a certain number of their resources need to be available for the application group to be available. The number of desired available resources is controlled by the *Availability Target* parameter in the customization dialog. You can use the preference value in the customization dialog to control which resources you want to be available and which you want to keep as backups.

The effective preference of each group member is calculated to determine which group members get picked to be activated by the group. The primary inputs into this are the base preference (as specified through the customization dialogs) and the preference adjustment (specified indirectly through the INGGROUP interface). These are added together to produce the resource's adjusted preference (this is the same as the preference value you actually enter the INGGROUP panels).

The following modifiers are then applied to the resource's adjusted preference if it is >1:

Bonus	Reason
25	Member is currently selected and has an adjusted preference ≤1000.
175	Member is currently selected and has an adjusted preference >1000.
220	Member has an observed status of Starting, Stopping or Degraded and the system is not being stopped.
225	Member has an observed status of Available and the system is not being stopped.

Bonus	Reason
–400	Member has an adjusted preference >1500 and either the system is being stopped or the system has already been stopped and the member thus has an observed status of SysGone.

The working value for the preference may then be set to an absolute value if any of the following conditions are satisfied:

1. A member with an effective preference of 1 may never be picked. It must always be sent a MakeUnavailable vote.
2. A member with an effective preference <600 may only be picked if the group is in recovery mode (that is, it is recovering from a failure).
3. A member requiring a resource on an excluded system must be sent a MakeUnavailable vote, even if it has been picked. This can occur with resources that have adjusted preferences >2600.

Available Preference Values

The preference values shown in the following tables are adjusted preference values (the sum of the base preference from the ACF and any preference adjustment that is specified through INGGROUP).

Table 6. Preference Values and Behavior of Members

Adjusted Preference	Behavior
3100	The resource will always be selected, even if it has an observed status of HardDown, or SysGone, its system is stopping or its system has been excluded.
2900	The resource will always be selected unless it has an observed status of SysGone or its system is stopping and there is a viable alternative resource. It will still be selected if its system has been excluded or it has an observed status of HardDown.
2400	The resource will always be selected unless it has an observed status of SysGone or HardDown, or its system is stopping and there is a viable alternative resource. It will still be selected if its system has been excluded.
1000	The member is always selected when the group has been started and the system associated to this member comes up. Other resources from the same group with preference 700 are stopped in favor of this member.
900	The member is always selected when the group has been started and the system associated to this member comes up. Members with preference 500 will be stopped in favor of this group.
700	The member is always selected when the group has been started and the system associated to this member comes up. Members with preference 300 will be stopped in favor of this group. It is not stopped unless a very high-preference alternative (1000) becomes available. This value is the default. Use it in a MOVE group for the resource that normally runs.
<600	This value is used as a threshold. A resource with preference 599 and lower will not be started at group startup, and is selected only if the group is in recovery mode.
500	This resource is not selected but will be selected when the group enters recovery mode. Once the problem is over, it will be stopped as soon as a primary member with a calculated preference >750 is available.

Table 6. Preference Values and Behavior of Members (continued)	
Adjusted Preference	**Behavior**
300	This resource is not selected but will be selected when the group enters recovery mode. Once the problem is over, it will be stopped as soon as a primary member with a calculated preference >550 is available.
100	This resource is not selected but will be selected when the group enters recovery mode. As soon as a preference 500 backup member is available, it will be stopped in favor of the backup member.
1	A resource with this preference will never be selected by SA z/OS automation; it is always sent an Offline vote. Operators may make such resources available interactively by using the INGGROUP command and giving it a higher preference. Use this value to provide completely manually controlled application availability.
0	This value indicates that the member is passive and will never have a vote propagated to it. If a member is made passive, any votes that have been propagated to it will be withdrawn.

Preference Scenarios

To understand the interactions of the various flavours of preference values and the concept of the recovery mode, the following sections describe some scenarios of how SA z/OS reacts according to the specified and automatically applied preferences.

Table 7. Preference Value Scenarios	
Scenario	**Description**
Backup with Non-Disruptive Restoration	In this scenario you have an application ABC that can run on one of two systems: SYS1 and SYS2. Only one instance of the application should normally be active. The instance on SYS1 is the primary instance (set preference 700), and the instance on SYS2 is the backup instance (set preference 500). The preferences 700 and 500 mean that the applications will behave as follows: 1. If SYS2 is IPLd first, the application will not be started (preference 500, not in recovery mode). When SYS1 is IPLd, its instance of the application will be started (preference 700, highest startable instance). 2. If SYS1 should fail, the instance on SYS2 will be started (preference 500, in recovery mode, highest startable instance). 3. When SYS1 is IPLd again, it will not start its instance (preference 700, but active instance with preference 750 (500+250 for being available). 4. At the end of the service period the instance of SYS2 will be stopped. 5. At the start of the next service period, the instance of SYS1 will be started (preference 700, highest startable instance).

Table 7. Preference Value Scenarios (continued)

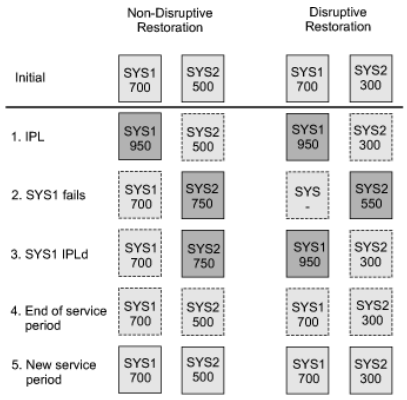
Scenario	Description
Backup with Disruptive Restoration	This scenario is like the above, but you want the application to return to SYS1 as soon as SYS1 is available, even if this causes an additional service interruption. You can use preferences of 700 for SYS1 and 300 for SYS2, or 900 for SYS1 and 500 for SYS2. As these are the only two applications involved, either combination will work (the important thing is that they are more then 250 apart). The example uses the 700 and 300 values. 1. If SYS2 is IPLd first, the application will not be started (preference 300, not in recovery mode). When SYS1 is IPLd, its instance of the application will be started (preference 700, highest startable instance). 2. If SYS1 should fail, the instance on SYS2 will be started (preference 300, in recovery mode, highest startable instance). 3. When SYS1 is IPLd again, its instance will be started (preference 700, highest startable instance) and the SYS2 instance will be stopped (preference 550 (300+250 for being available), lower than 700). This causes a short service interruption as the application moves back to SYS1. 4. At the end of the service period the instance of SYS1 will be stopped. 5. At the start of the next service period, the instance of SYS1 will be started (preference 700, highest startable instance). Figure 14 on page 57 illustrates the difference of non-disruptive and disruptive restoration between the steps 2 and 3 in each scenario: in the non-disruptive scenario, the application ABC continues to run on SYS2 until the end of the service period for the group, because this resource now has a higher priority of 750 than the resource on SYS1, which would only have priority 700. In the disruptive restoration on the other hand, SYS1 takes over for the rest of the service period after the problem has been fixed.  *Figure 14. Non-Disruptive and Disruptive Restoration*

Table 7. Preference Value Scenarios (continued)

Scenario	Description
Real Hierarchy of Preferences	In this scenario, you have three systems where the application can be run. SYS1 is where you normally run it, SYS2 is a back up and SYS3 is where it can run in extreme emergency only. You would specify preferences as 700, 500 and 100. 1. If either SYS2 or SYS3 are IPLd first, their instances will not be started (preferences 500 and 100, not in recovery mode). The operator can, of course, explicitly request one of the instances to be started (which should be done if they want the application to be available and know that SYS1 will not be IPLd for a while). 2. When SYS1 comes up its instance will be started (preference 700, highest startable instance). 3. If SYS1 fails, the SYS2 instance will be started (preference 500, in recovery mode, highest startable instance). 4. If SYS1 were to return at this point, we would be back in the first scenario, a non-disruptive backup. 5. Instead, let us suppose that SYS2 now fails. This causes the instance on SYS3 to be started (preference 100, in recovery mode, highest startable instance). 6. If either of the other two systems came back (let us assume it is SYS2), its instance will be started (preference 500, in recovery mode, highest preference instance) and the instance on SYS3 will be stopped (preference 350 [100 plus 250 for being available], lower than 500). This would cause an application outage, but would get the application off SYS3. 7. If SYS1 then comes back, its instance will not be started: Preference 700, lower than 750 (500+250 for being available). 8. After the next service period stop, we would be back to running the instance on SYS1. **Note:** If SYS1 was down at the end of the start of the next service period, the application would not be restarted. Using preferences of 900, 700, 100 (or 300) would get around this, but also means that the application could be automatically started on SYS2 if it was IPLd before SYS1.

Table 7. Preference Value Scenarios (continued)

Scenario	Description
Server Battery with Backups	In this scenario your sysplex consists of four systems (SYS1, SYS2, SYS3 and SYS4). You defined a Sysplex Application Group of nature SERVER with an availability target of 3 that should normally run on systems SYS1, SYS2 and SYS3. A backup resource is defined on SYS4, but it should only be used when one or more of the main servers is unavailable. A preference of 700 should be used for the resources on SYS1, SYS2 and SYS3, and the backup resource on SYS4 should have a preference of 300. 1. When all systems are IPLd, the resources on SYS1, SYS2 and SYS3 will be started (preference 700, 3 highest startable resources). The resource on SYS4 will not be started (preference 300, not in recovery mode). 2. If either SYS1, SYS2 or SYS3 fails, the resource on SYS4 will be started (preference 300, now in recovery mode). 3. When the failed system is restored, its resources will be restarted (preference 700, three highest startable resources) and the resource on SYS4 will be stopped (preference 550 [300+250 for being available]). **Note:** Setting a preference of 500 for the resource on SYS4 would cause it remain running when the failed system returned, effectively turning the resource on the returned system into the backup. It is also possible to set the preferences as 700, 700, 700, 690, which will let automation start the resources on whichever systems it needs to make the application available, but to prefer not to use the SYS4 resource if the other three are available (which they may not be at IPL time).
Move Groups and Recovery Mode	In this scenario your sysplex consists of four systems (SYS1, SYS2, SYS3 and SYS4). You defined a Sysplex Application Group of nature MOVE (availability target of 1) that should normally run on system SYS1 (that is, preferences of 700 for SYS1, 450 for SYS2, 250 for SYS3, and 1 for SYS4). 1. When all systems are IPLd, the resource on SYS1 will be started (preference 700, highest startable resource). The resources on SYS2, SYS3, and SYS4 will not be started (availability target of 1 and APG not in recovery mode with preferences ≤600). 2. If SYS1 should fail then we enter recovery mode, which means that SYS2 and SYS3 become eligible to be started. In this case SYS2 will be chosen (highest eligible preference 450). If however, SYS1 was a planned shutdown (INGREQ ALL) then we would not be in recovery mode and neither SYS2 nor SYS3 would be eligible, so the application would not be moved automatically. To move the resource to another system use INGGROUP to raise that system's preference to >600. 3. If SYS2 should fail, as well as SYS1, the resource will be started on SYS3 without further adjustment (minimum startable preference 250). **Note:** SYS4 has not participated in any of these MOVE examples (preference value 1, non-startable instance).
Absolute manual control	Member to be activated gets preference 3200, all others get preference 1. The member with the 3200 preference will be selected. All others will always have a MakeUnavailable vote propagated to them. This does not change even if the system with the selected member leaves the sysplex. A shutdown of the primary system will not initiate a move of the application, as the 400 point penalty is not sufficient to drop its preference below 2600.

Table 7. Preference Value Scenarios (continued)

Scenario	Description
Move only after system failure	Primary member defined preference 2800, backups preference 500. As long as the primary is not SYSGONE it will be selected. If the primary is SYSGONE, its preference drops to 2575 (2800 – 400 + 175), allowing an alternative viable member to be selected. The backups will only be viable if the resource was previously active because of the groups recovery mode. A shutdown of the primary system will not initiate a move of the application. Although the primary systems preference will be below 2600, there will be no viable alternatives as the backups may only be used if the group is in recovery mode.
Move only if system down or stopping	Primary member defined preference 2800, backups preference 700. As long as the primary member is not SYSGONE and its system is not being stopped it will be selected. If the primary member is SYSGONE or its system is being stopped, its preference drops to 2575 (2800 – 400 + 175), allowing an alternative viable resource to be selected. The backups are always viable. A shutdown of the primary system will initiate a move of the application during the shutdown.
Sticky move only after system down or stopping	Primary member defined preference 2801, backups preference 2800. As long as the primary member is not SYSGONE and its system is not being stopped, it will be selected. If the primary becomes SYSGONE or its system is being stopped, its preference drops to 2576 (2801 – 400 + 175), allowing one of the backups to be selected. The sticky bonus raises its preference to 2975 (2800 + 175), meaning that even when the primary returns the selected backup system will remain selected. It will only be deselected if it goes to SYSGONE or its system is being stopped, in which case the primary will be re-selected if it is available otherwise another backup would be picked. A shutdown of the primary system will initiate a move of the application during the shutdown.
Move only after system or application failure	Primary member defined with preference 2400, backups preference 500. As long as the primary member does not fail to SYSGONE or HARDDOWN it remains selected. If it gets to HARDDOWN or SYSGONE without failing, there are no viable alternatives, so it will not be moved. If it does fail, its preference is between 2600 and 2000 and the backups are viable, so one of them can be picked. The backups are only viable if the group is in recovery mode (that is, the member was active when the failure occurred).

Table 7. Preference Value Scenarios (continued)

Scenario	Description
Move only if system down, being stopped or application failure	Primary member defined with preference 2400, backups preference 700. As long as the primary member is not SYSGONE or HARDDOWN and its system is not being stopped it remains selected. If it is HARDDOWN or SYSGONE or its system is being stopped its preference is between 2600 and 2000, so a viable alternative can be selected. A shutdown of the primary system will initiate a move of the application, as it will have lost its 225 point bonus for being available and have incurred a 400 point penalty, bringing its preference down to well below 2600, allowing the selection of an alternative resource if there is a viable one.
Sticky move if system down or being stopped, move after application failure	Primary member defined with preference 2401, backups preference 2400. As long as the primary is not SYSGONE or HARDDOWN and its system is not being stopped it remains selected. If it is SYSGONE or HARDDOWN or its system is being stopped its preference is between 2000–2600, so a viable alternative may be chosen. The sticky bonus raises the preference of the selected member to 2575 (2400 + 175), so it will remain selected until it gets to a state of SYSGONE or HARDDOWN or its system is being stopped, at which point another viable alternative will be selected, with the primary member having preference. The individual shutdown of the active member will not trigger a move. A shutdown of the primary system will result in a move of the application. It will have a 400 point penalty, dropping its preference to between 2600 and 2000, permitting the selection of a viable alternative.
Sticky move group, unless member stopped	Primary member defined with preference 701, backups preference 700. As long as the primary member is startable, it remains selected. If it becomes non-startable, another member is selected and that remains selected as long as it is startable. Individually stopping the active member will trigger a move. If the members were defined with preferences >1500, the bonus would be 175 points rather than 25 points.
Sticky move group, until told otherwise	Primary member defined with preference 3200, backups with preference 1600. With base preferences of 3200, you get it picking a member when the sysplex is IPLd and retaining that one until an operator uses INGGROUP to make it pick another one. Beware of timing problems as the first system with a member to register will probably get it.

SA z/OS Automation Flags

Automation flags are automation policy settings defined in the customization dialog that specify whether certain types of automation are on or off for a resource.

The Automation Flag Specification panels of the customization dialog define the cumulative effects of your automation flag settings, including:

- When automation is active
- Resources that automation applies to

- The types of automation mode you want SA z/OS to handle, such as starting and shutting down applications
- Any exceptions to the defined automation
- Whether any automated operator function will operate in *log mode*, an SA z/OS option allowing a human operator to log the flow of an automated operation

SA z/OS automation flags control whether automation occurs for a resource and, if so, when. Using the customization dialog, you can set default automation flags for all systems and applications, or you can override automation flags for specific systems or applications.

See also "Determining Automation Agent Flag Values" on page 135.

Automation Agent Flags

If an event occurs that triggers automation, SA z/OS checks the appropriate flag to determine whether automation is currently on.

If it is, SA z/OS proceeds with the automation actions that are associated with the flag. For example, if TSO fails and the Restart flag is on for TSO, SA z/OS restarts TSO according to the policy that is specified in the customization dialog.

Figure 15 on page 62 shows how specific automation flags cover the phases in the lifetime of an application.

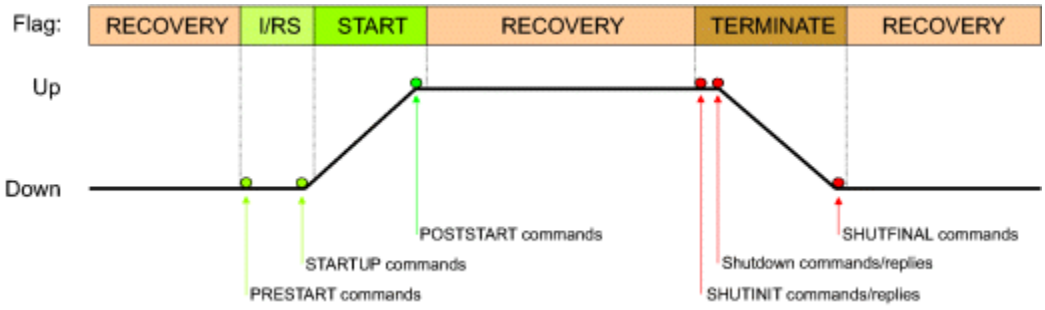


Figure 15. Automation Flag Span of Control

PRESTART and STARTUP commands are only issued if the INITSTART respective RESTART flag is turned on. POSTSTART commands are only issued if the START flag is turned on. SHUTINIT, SHUTNORM, SHUTIMMED, SHUTFORCE and SHUTFINAL commands are only issued if the TERMINATE flag is turned on.

REFRESHSTART and ANYSTART commands are issued regardless of any automation flag specification in case of NetView recycle or INGAMS REFRESH processing. During the start-up process of an application the ANYSTART commands are only issued if the START flag is turned on.

There is one flag that controls all the other flags. This is the Automation flag:

Automation

If this flag is set to Y (Yes), SA z/OS checks each of the other flags to see if they are set to Y or N.

If this flag is set to N (No) or S (Suspended) , no automation is allowed, regardless of how the other flags are set. By default, message capturing, threshold checking, and WLM resource management are still done, even when this flag is set to N or S. You can control these behaviors in the Customization Dialog > System Defaults > AUTOMATION OPTIONS policy. For more details, see "System Defaults Entry Type" in *IBM Z System Automation Defining Automation Policy*.

Note: The value of S (Suspended) can occur only if a resource is suspended using the INGSUSPD command or if it's part of the suspend file. It is neither possible to set the agent automation flag manually to S nor defining the value S in the customization dialog.

The other flags that apply to each resource are:

Initstart
If this flag is turned on and the SA z/OS automation status for a subsystem is DOWN, SA z/OS can try and start the subsystem when all other conditions necessary for its startup (as defined in your automation policy) are met.

Start
If this is on, automation manages and completes the startup process of the subsystem.

Recovery
If this is on, automation takes the specified action for any MVS message that is not associated with the initialization, startup, shutdown, or restart of a subsystem. For example, if this flag is on, automation takes corrective actions in response to messages that indicate shortages of WTO buffers and JES spool space. Automation also invokes offload routines when the following MVS data sets are about to fill up: SYSLOG, LOGREC, SMF, MVS dump.

Terminate
If this flag is turned on, SA z/OS can process automated shutdown requests for this subsystem. The setting of this flag can be overridden when the shutdown is requested. Also, the flag can be used to control secondary automation that occurs during the shutdown process. Secondary automation covers, for example, replying asynchronously to WTORs and issuing commands in response to messages from the subsystem.

Restart
If this flag is turned on and the SA z/OS automation status for a subsystem is RESTART, SA z/OS can try and start the subsystem when all other conditions necessary for its startup (as defined in your automation policy) are met.

When you set an automation flag to Y in the customization dialog, that automation flag is on all the time, unless someone does one of the following:

- Turns the flag off for a period by scheduling an override in the customization dialog. For example, you could schedule an override that turns initialization and shutdown automation off on Sundays from 12:00 to 16:00, so you can do your system backups.

  You can display these scheduled overrides using the DISPSCHD command dialog, as explained in “Displaying Automation Agent Flag Override Settings” on page 136.
- Turns the automation off using the INGAUTO command dialog. See “Setting Automation On or Off: INGAUTO” on page 137.
- Causes a higher level flag to be turned off, for example the subsystem's Automation flag or the SUBSYSTEM Restart flag.
- Issues a suspend request against the affected resource using the INGSUSPD command dialog (see INGSUSPD) or define the affected resource in the suspend file (see “Using the Suspend File” on page 89).

If a flag is set to N, you cannot schedule an override in the customization dialog to set it on for a particular time. You must use the INGAUTO command dialog.

If a flag is set to S, you cannot schedule an override in the customization dialog to set it on for a particular time. At first you have to resume the affected resource using INGSUSPD or INGVOTE.

If a flag is set to E, an exit is executed to determine whether automation is allowed. See *IBM Z System Automation Customizing and Programming* for details of Flag Exits.

If a flag is set to L, triggered automation actions such as commands or replies that this automation flag is checked for by SA z/OS are written to the NetView Log instead of being issued.

Note: When the automation agent receives a start or stop order from the automation manager, the agent will check the relevant automation agent flag before issuing the actual start or stop command for the resource. If automation is not allowed according to the flag setting, the automation agent will reschedule itself on a timer and re-check the flag setting.

For a start request, the resource's Start Delay time is used for the timer rescheduling. For a stop request, the resource's Shutdown Pass Interval is used for the timer rescheduling. It means that turning the relevant automation flag on while the automation agent has rescheduled itself will not immediately result

in starting or stopping of the resource. The resource's start or stop commands will be issued only after the relevant timer expires.

Setting Automation Agent Flags

You can set automation agent flags for:

- System Defaults (using their AUTOMATION FLAGS policy item)
- Application Defaults (using their AUTOMATION FLAGS policy item)
- MVSCOMP Defaults (using their AUTOMATION FLAGS policy item)
- Applications (using their AUTOMATION FLAGS or MINOR RESOURCES policy item)
- MVS Components (using their MINOR RESOURCES policy item)

Note: For MVS components, only the Automation and Recovery flags apply.

Because flags have inherited values, it is not necessary to define them all explicitly. Specific resources need only have flags defined if they differ from the values inherited from the defaults.

A common step during automated handling of events is to check the automation flags to determine whether automation is on or off and whether automated handling of an event can continue.

The AOCQRY command performs this automation flag check. See *IBM Z System Automation Programmer's Reference* for more information.

The benefit of this automation hierarchy is that you can customize SA z/OS to automate as many activities and resources as necessary in the z/OS environment. You can choose to have SA z/OS handle all phases of startup, monitoring, recovery, and shutdown for all resources or have a human operator control some phases for some resources. You can control all automation by setting just one flag (the Automation flag) on one panel and then setting exceptions. The simplest way to automate your resources is to use the Automation flag.

Periods that automation is not to apply for can be specified.

There are several commands to display automation flag settings, for example, DISPFLGS. The INGAUTO command dialog can be used to set or override automation flags, if the affected resources are not suspended.

SA z/OS provides an exit capability for automation flag processing. See *IBM Z System Automation Customizing and Programming* for more information.

Relationship Between Automation Agent Flags and Resource Status

From initialization through normal operation to shutdown, a resource can have several automation modes, such as being started, in an up state, abending, and shutting down.

SA z/OS uses *resource states* to track these automation modes for monitored resources. A resource must have a particular status for certain automated actions to occur. The effect of these actions may, in turn, change the resource status from one value to another.

Understanding how SA z/OS typically uses resource states and automation agent flags is important when planning and writing your own automation procedures. For more details on automation agent flags, resource states, and their use in automation procedures, see *IBM Z System Automation Customizing and Programming*.

SA z/OS provides an exit for processing state changes. See *IBM Z System Automation Customizing and Programming* for more information.

Resource States and the Status Display Facility

SA z/OS also uses resource states in status information forwarded to the status display facility (SDF).

This status information results in status changes being reflected on SDF panels.

Automation Agent Flags and Minor Resource Flags

In addition to the automation agent flag settings for the entire application, you can set flags for minor resources (MINOR RESOURCES policy item) that control responses to specific situations.

Typical minor resources include message IDs and states.

For example, TSO is defined as application TSO. You have defined a reply to message IKT010D on the MESSAGES/USER DATA policy item of the Application policy object for TSO. You can change the shutdown process by:

- Turning the Terminate flag for TSO off. This means you have to shut down TSO manually.
- Turning the Terminate flag for minor resource IKT010D off. This allows SA z/OS to shut down TSO normally. When the IKT010D message is issued, shutdown processing waits for a manual response.

Automation Manager Flags

Global Automation Manager Flag

Using the INGLIST or the INGSET command (see *IBM Z System Automation Operator's Commands*), you can set an automation flag for the individual resources, which is checked by the automation manager before it sends any order to the automation agent to start or stop the specific resource.

The purpose of this flag is to prevent (if flag is set to NO) or enable (if flag is set to YES) the starting or stopping of resources from the point of view of the automation manager. This can be done for resources that reside on systems that are currently inactive, for example, to prevent the startup of the resource at IPL time of the system.

Automation Manager Suspend Flag

The automation manager suspend flag is set with the INGSUSPD command (see *IBM Z System Automation Operator's Commands*) or during processing of the suspend file (see “Using the Suspend File” on page 89). This flag is also checked by the automation manager before it sends any order to the automation agent to start or stop the specific resource.

If the flag has the values DIR (Direct) or IND (Indirect), the manager is not going to start or stop the resource. To understand the differences between this flag and the other automation flags described before, see “Suspend and Resume Functionality” on page 85.

Possible values for the automation manager suspend flag are:

DIR
The resource is directly suspended (got a suspend request) via INGSUSPD or the suspend file.

IND
The resource is indirectly suspended (because a supporting resource was suspended) via INGSUSPD or the suspend file.

PEN
The suspend request has been recognized but the resource is currently in a start or stop process. Therefore, the request is held back until the desired status has been reached.

Note: As long as the suspend request on a resource is pending, the agent doesn't get a suspend order and the agent automation flags do not display an 'S' for the automation agent flag value.

UNSUSPENDED/blank
The resource is not suspended. In INGINFO, you see the value of UNSUSPENDED behind the manager suspend flag, but in INGLIST this is only shown as a blank field.

Statuses Supplied by the Automation Manager

The automation manager keeps track of various statuses for each resource:

Observed Status
This indicates the current status of the resource, as reported by the automation agent. See “Observed Status” on page 67.

Automation Status
The automation status of a resource tells the automation manager what automation status a resource is currently in.

If a resource's automation status is currently *idle*, the automation agent does nothing for automation. Conversely, if the automation status is *denied*, automation is currently not allowed, possibly because the automation flag has been set off. See “Automation Status” on page 68.

Desired Status
This status reflects the goal of a resource. The automation manager will be continuously trying to place the resource into this status unless the goal is revoked or changed. This status can be either *Available* or *Unavailable*.

If it is *Available*, the automation manager will try to start the resource. If it is *Unavailable*, the automation manager will try to stop the resource.

You can define a *default* desired status for a resource:

ALWAYS
The desired status is set to Available and the resource is started once its dependencies are satisfied.

ONDEMAND
The desired status is set to Unavailable and the resource is not be started. To start it a MakeAvailable vote must be propagated to it. This can come from an operator request, a service period, or an application group request.

ASIS
The desired status is always set to the observed status, that is, the group remains in the status it currently has and no action is taken by SA z/OS at any time, as long as there is no request placed or propagated to the group.

See “Desired Status” on page 69.

Startability Status
This is a calculation as to whether it is possible to start the resource if this is requested. It includes the startability of all inactive start dependent resources. This status is mainly used as input for Compound Status calculations and by groups to prevent them from selecting members that cannot be started. See “Startability Status” on page 69.

Compound Status
This status is the composition of the other statuses to give the operator a chance to check from the automation manager if a resource is currently OK concerning the automation. See “Compound Status” on page 69.

Health Status
The health status can be used by the automation manager to make decisions and, if necessary, trigger automation for the application if there is a change in the health state. See “Health Status” on page 70.

An operator can retrieve the status using two commands:

- INGLIST provides some main attributes of a resource, for example, a resource's different statuses.
- INGINFO provides information about a resource from the point of view of the automation manager.

Three of these statuses are supplied by the automation agent that is responsible for running the resource:

- Observed status

- Automation status
- Health status

The desired status is derived through the request propagation mechanism. The startability status is derived through status propagation.

The compound status is a summary of all of the other statuses and a number of other values. The compound status is displayed with INGLIST or INGINFO.

Observed Status

This represents the current status of the resource and is provided by the related automation agent.

Possible values are:

Unknown (Assumed unavailable)
The automation manager has no observed status information about the resource. To be on the safe side, the automation manager will assume that the resource is unavailable and will try to make it available if this is the desired status. On the other hand, this assumption allows the automation manager to process shutdown requests for supporting resources that can only be made unavailable if this resource is unavailable.

SysGone (unavailable)
The system that the resource is defined on is no longer a member of the sysplex.

SoftDown (unavailable)
The resource is unavailable and automation may restart it.

HardDown (unavailable)
The resource is unavailable and automation will not restart it.

Starting (available)
The automation agent is either in the process of making the resource available or has detected actions from some outside command to make the resource available.

Available
The resource is ready for use.

Degraded (available)
The resource is available but not all required members in the server group are available.

Standby (unavailable)
The resource has a primary/secondary system association defined. The automation agent posts this status to the automation manager when setting the automation agent status to MOVED or FALLBACK.

Note: The automation manager treats STANDBY like a HARDDOWN status except that it is not considered that this indicates an error condition.

Problem (available)
The resource is available but has a serious problem, probably making it unusable.

Stopping (available)
Either the automation agent is in the process of stopping the resource, or it has detected that some outside command is doing so.

WasAvailable (Assumed available)
The automation manager has lost contact with the automation agent responsible for the resource, however the resource was available when contact was lost and the system it is on still appears to be running. The resource will be treated as being available. This is primarily to prevent inappropriate recovery actions from being initiated whenever an SA z/OS NetView is recycled.

Observed Status Transitions

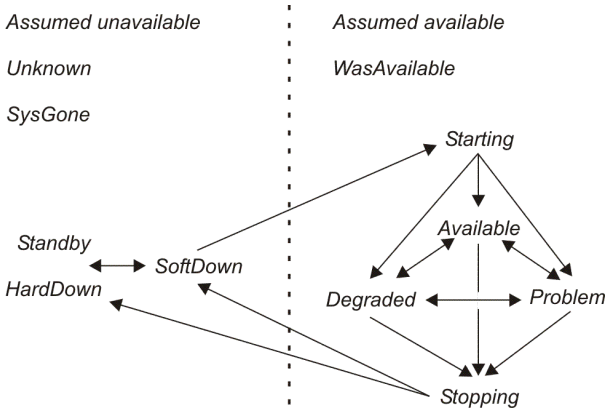


Figure 16. Observed Status Transitions

Automation Status

This indicates the status of the automation for the resource. It is provided by the related automation agent.

Possible values are:

Unknown
No connection with the automation agent.

Idle
No orders have been sent; the automation agent is not doing anything. The manager may send the automation agent new orders.

Ordered
Orders have been sent. This status is only posted within the automation manager.

Busy
The automation agent is processing the orders, or observing/assisting a start or stop process initiated by some outside command. This status is entered when a resources status becomes Starting or Stopping and the previous Automation Status was Idle or Ordered. This status is changed to Idle if the resource achieves an observed status other than Starting or Stopping. This status may be posted directly by the automation agent.

Denied
The automation agent was not allowed to process the last order it received. This status is changed to Idle if the resource achieves an observed status of HardDown, SoftDown or Available. This status is posted directly by the automation agent.

Problem
The automation agent encountered a problem while carrying out processing for the resource. This status is changed to Idle if the resource achieves an observed status of HardDown, SoftDown or Available. This status is posted directly by the automation agent.

Internal
This means that the automation of the resource is being handled internally.

Automation Status Transition

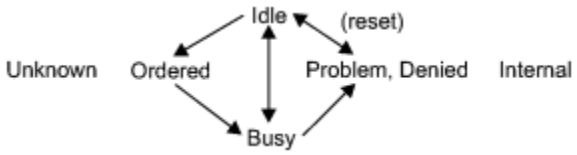


Figure 17. Automation Status Transitions

The automation manager may only send orders to an idle automation agent.

Transition from Idle to Busy occurs when the automation agent detects a third party (such as an operator or ARM) starting or stopping a resource.

Desired Status

You can set the desired status of a resource either:

- With the INGREQ command
- With a service period
- With a schedule or resource override (INGSCHED)
- Or indirectly through a vote that stems from the request propagation

Possible values are:

Available
The resource should be started (made available).

Unavailable
The resource should be stopped (made unavailable).

Startability Status

The startability status is calculated from the resource's current status, its automation status, its automation flag and the startability of its supporting resources.

Possible values are:

Yes
The resource is either startable or already started.

No
The resource cannot be started due to a problem with the resource itself.

Inhibited
The resource cannot be started because of either a problem with one of its supporting resources or because automation has been prohibited.

Denied
The resource cannot be started because the automation status of this resource is *Denied*.

Compound Status

This status is a summary of all the statuses of a resource. It considers elements of the four other statuses as well as a number of other values.

Possible values are:

Problem
There is a problem with this resource that automation cannot solve. Operator intervention is required.

Denied
The resource is not in its desired state and automation is unable to proceed because either the automation flag has been turned off or the automation manager hold flag for the resource is set to YES.

Inhibited
The resource is not in its desired state and automation is unable to proceed because of a problem with a supporting resource. Operators should try to fix supporting resources with compound status *Problem* and *Denied*.

Awaiting [Automation]
The resource is not in its desired status and SA z/OS is waiting for its supporting resources to reach the appropriate state.

InAuto[mation]
SA z/OS is in the process of starting or stopping the resource.

Degraded
For a group it means that it is partially running, but not at full capacity. For normal resources, it can mean that the resource is Starting or Stopping, or that the application is suffering from a performance or throughput problem (corresponding to automation agent status HALTED).

Satisfactory
The resource's desired and observed statuses are corresponding; no further automation or operator activity is required.

Health Status

With application-specific performance and health monitoring, a separate status shows up to inform you about the application's health.

Possible values are:

Normal
The observed resource is behaving normally.

Warning
The resource's health state has crossed the warning threshold.

Minor
The resource's health state has crossed the minor threshold.

Critical
The resource's health state has crossed the critical threshold.

Fatal
The resource's health state has crossed the fatal threshold.

Unknown
The health state of the resource is not known.

Application Statuses Supplied by Automation Agents

There are different types of statuses that may be applied to an application. They are all retrieved on the side of the automation agent.

These are:

- “Automation Agent Statuses” on page 70
- “Application Monitor Status” on page 78
- “Automatic Restart Manager Statuses” on page 78

This chapter discusses the following related topics:

- “Mapping the Automation Agent Statuses to the Automation Manager Observed Status” on page 79
- “SA z/OS Processes Involved in Status Determination” on page 83

Automation Agent Statuses

SA z/OS defines different automation agent statuses that it applies to automated resources. The automation agent status of a resource is determined from a combination of its application monitor status, its desired status, recent history, and intended action.

These statuses are used on the automation agents to control the automation. They are translated to *observed statuses* (see “Mapping the Automation Agent Statuses to the Automation Manager Observed Status” on page 79) and then sent to the automation manager.

On the automation agent you can use the DISPSTAT and DISPINFO commands to see the automation statuses.

Table 8 on page 71 gives a brief description of each automation agent status.

Table 8. Automation Agent Statuses

Value	Meaning
ABENDING	The application is undergoing a recoverable abnormal end. This status is entered when a TERMINATION message with the ABEND=YES attribute is received. It remains in ABENDING until its final termination message is received. If the application is defined to Automatic Restart Manager it may be posted to EXTSTART when Automatic Restart Manager attempts to recover it. If the application has not exceeded its critical threshold, SA z/OS posts the application to RESTART and attempts to restart it. If it has exceeded its critical threshold it is put into BROKEN status when it has been cleaned up, and SA z/OS does not attempt to start it. If it is slow to leave the system after its FINAL message is received it may go to ZOMBIE status. If the application is undergoing a normal shutdown by SA z/OS no further shutdown commands are issued. The shutdown may be resumed by using the INGREQ command with a stronger type.
ACTIVE	The application is running, but is not yet ready for work. An application can be put into ACTIVE status in response to the following conditions: • SA z/OS has received an ACTIVE message for the application. • The SA z/OS startup checker has run for the application when its automation status was STARTED. The startup checker found the application monitor status for the application to be ACTIVE. • During routine status checking the application monitor status for the application was found to be ACTIVE when its automation status indicated it should be otherwise. • SA z/OS attempted to start the application, but found that its application monitor status was ACTIVE. • SA z/OS found that the Automatic Restart Manager status for the application is STARTING, but has not received any messages concerning the application status. • SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED, but the application monitor status for the application is ACTIVE. An application remains in ACTIVE status until its UP message is received. If the application is starting or restarting, an SA z/OS monitoring check for the application is done after its start delay time and number of start cycles. If the application is not found and if the critical threshold is not reached, SA z/OS will attempt to restart the application. If it is found, but is not yet UP, it is put into STARTED2 status.
AUTODOWN	The application is shut down. SA z/OS may restart it to comply with an operator request. If the shutdown specified that the application was to be restarted, it is put into RESTART status when the shutdown is complete. 1. SA z/OS has shut the application down. 2. When SA z/OS initialized, the operator replied NOSTART to the AOF603D WTOR. Any applications that would have been put into the DOWN status during initial status determination have instead been put into the AUTODOWN automation status. You can use the SETSTATE command to change the application status to either RESTART or CTLDOWN. SA z/OS may attempt to restart an application in AUTODOWN status when SA z/OS is reloaded, or when an operator requests SA z/OS to start one of the application descendents. In both cases, the application goes to RESTART status.

Table 8. Automation Agent Statuses (continued)

Value	Meaning
AUTOTERM	SA z/OS is in the process of shutting the application down. The shutdown is in response to a INGREQ REQ=STOP command. This status persists until SA z/OS is sure that the application has been cleaned up. Many things may happen to an application that is being shut down. If the shutdown is successful, the application is placed in either AUTODOWN or CTLDOWN status. If the shutdown specified that the application should be restarted it goes through AUTODOWN to RESTART status. If the application abnormally ends while it is being shut down it may go into either ABENDING or BREAKING. A normal shutdown will stop processing an application that abends, but other shutdowns will continue. If the shutdown runs out of commands to issue, the application is placed into STUCK status. If it has problems shutting down, the application may be placed into ZOMBIE status.
BREAKING	The application is undergoing a nonrecoverable abend; that is, it has received a termination message specifying BREAK=YES. If the application is undergoing a normal shutdown by SA z/OS no further shutdown commands are issued. The shutdown may be resumed by using the INGREQ REQ=STOP command. This status persists until SA z/OS receives its final termination message and is sure that the application has been cleaned up. If the termination experiences difficulties, the application may be posted to ZOMBIE status.
BROKEN	The application has suffered a nonrecoverable abend. SA z/OS will not restart it. An application can be put into BROKEN status in response to the following conditions: 1. The application has suffered a nonrecoverable abend, indicated by the reception of a TERMINATION message with the BREAK=YES attribute. 2. The application has suffered sufficient recoverable abends to exceed its critical threshold. 3. Issuing the application's prestart or startup command resulted in a non-zero return code. This status is preserved across a recycle of SA z/OS or a re-IPL of the processor, unless the application does not have its Restart after IPL option set to NOSTART.
CTLDOWN	The application is shut down and SA z/OS is not allowed to restart it. 1. An operator asked SA z/OS to shut the application down and not to restart it until authorized to do so by an operator. 2. An operator used a SETSTATE command to tell SA z/OS that an application should not be restarted until an operator authorizes SA z/OS to do so. You can use the SETSTATE command to change an application status from CTLDOWN to RESTART or AUTODOWN, in which case SA z/OS will attempt to restart it.
DOWN	The application has not been started during the lifetime of this SA z/OS. The DOWN status is set only during initial status determination and is possible only if the application monitor status is INACTIVE. The automation status of the application when SA z/OS was last shut down on this system is used in the following manner to determine if it is to be placed into the DOWN status. 1. The previous automation status was not one of STOPPED, CTLDOWN or BROKEN. 2. The application does not have its Restart after IPL option set to NOSTART, the previous state was BROKEN, CTLDOWN or STOPPED, and this is the first time the agent has been started since the system was IPLed.

Table 8. Automation Agent Statuses (continued)

Value	Meaning
ENDED	This status is used for transient applications only, and indicates that the job for the application has finished and left the system without any errors. Any start-dependent resources for the application will be started as though it were a normal z/OS subsystem that was UP. If the transient application can be rerun, you can use the SETSTATE command to restart it. If the transient application cannot be rerun, it will remain in ENDED status.
ENDING	A transient application is in the process of terminating. A transient application goes to ENDING status when a termination message is received for it, and it is not being shut down by SA z/OS. This status shows that the application is terminating, but that this is expected. A transient application may also go to ENDING if an operator is shutting it down outside SA z/OS control, or if it has abnormally ended, but the abend messages are being treated as normal termination messages. The application remains in ENDING status until either: • An abend message is received that will put it into either ABENDING or BREAKING status. If the application abends then either SA z/OS or Automatic Restart Manager can restart it. • The application final termination message is received, at which point the RESTARTOPT for the application is checked. If it is ALWAYS then the application is put into RESTART status and SA z/OS will attempt to restart it. If it is anything else, the application goes to ENDED status. It is assumed that if a transient application ends normally then it will deregister from Automatic Restart Manager. If it is slow to clear the system after its FINAL message is received it may go to ZOMBIE status. If an ACTIVE or UP message is received for the application, its automation status is changed to either ACTIVE or UP, as appropriate.
EXTSTART	SA z/OS has determined that the application is being started or restarted by an agent external to SA z/OS. In situations where SA z/OS is able to identify the external agent (such as Automatic Restart Manager), it takes appropriate steps to monitor that agent's actions and, if necessary, step in to assist it. An application can be put into EXTSTART status in response to the following conditions: • SA z/OS is unable to identify the external agent. • Automatic Restart Manager is in the process of restarting the application.
FALLBACK	The application is not running on the primary system where it should run, but this status has been encountered for this application on one of its secondary systems. It should be active on another system. If the other system fails, the application can fall back to this system, where it could possibly be restarted. However, SA z/OS, will not perform the restart on the fallback system, but this may be done by an operator request. This is implemented to leave the decision of restarting the application on the fallback system to the operator. An application can be put into FALLBACK status in response to the following conditions: • It is defined with a secondary association on this system. • An operator has used the SETSTATE command to put the application into the MOVED status. If this is one of the secondary systems for the application it will go to FALLBACK instead.
HALFDOWN	SA z/OS was in the process of shutting the application down, but the stop request was canceled while it was in progress. The application shutdown did not complete (for example, ASCBs may still be active). You may sometimes find that some, but not all, of the shutdown commands have been issued. To recover an application from HALFDOWN status you must determine where it is in its shutdown process and complete the shutdown manually. Applications go into HALFDOWN only when you cancel a stop request. Alternatively, you can use SETSTATE to put the application back into UP or RUNNING status.

Table 8. Automation Agent Statuses (continued)

Value	Meaning
HALTED	The application is still running, but something has happened that may have severely impacted its capabilities. 1. SA z/OS has received a HALT message for the application. 2. SA z/OS has detected that the application represents a JES2 application that is running short of spool space. 3. The Automatic Restart Manager status for the application is ELSEWHERE, but SA z/OS found its application monitor status to be either STARTING or ACTIVE. An application is taken out of HALTED status if its UP message is received. Also, operators may use the SETSTATE command to put the application into UP or RUNNING status.
MOVED	The application is not running: this is one of its primary systems: it should be active on another system. An application can be put into MOVED status in response to the following conditions: • An operator has used the SETSTATE command to put the application into the MOVED status. This is possible only on a primary system. • The startup detected that Automatic Restart Manager has started the subsystem on another system. A subsystem will remain in the MOVED status until it is restarted on the primary system by an external agent, such as an operator.
RESTART	The application is ready to be started. It has been previously active in the system. An application can be put into RESTART status in response to the following conditions: • The application abended and, after checking thresholds, SA z/OS is allowed to restart it. • SA z/OS has shut the application down in response to an operator request and is now preparing to restart it. • An operator has used the INGREQ REQ=START command to ask SA z/OS to restart the application. • SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED and the application monitor status for the application is INACTIVE. This implies that the attempt by Automatic Restart Manager to restart the application timed out while the application was in RESTARTING status. SA z/OS changes the automation status of the application to RESTART and attempts to start the application itself. During restart processing, the application RESTART automation flag is checked. If it is turned on, the application start commands are issued and the application is put into STARTED status. If the RESTART automation flag is off, the application remains in RESTART status and the startup monitor cycle initiates the startup process each time it runs.
RUNNING	This status is equivalent to UP, but is used for *transient* applications. It indicates that the UP message has been received for the transient application, or an operator has used the SETSTATE command to change the status of a transient application to UP. A transient application is one that SA z/OS expects to terminate on its own. When the job finishes the application goes through ENDING status to ENDED, at which point its descendants are started. Unlike the UP status, the descendants of a transient application are not started until it has terminated. A transient application should leave RUNNING status on its own. If it gets stuck, you should investigate it. You can use the INGREQ REQ=STOP command to put it into AUTODOWN status.

Table 8. Automation Agent Statuses (continued)

Value	Meaning
STARTED	The commands to start the application have been issued, but it has yet to start running. An application can be put into STARTED status in response to the following conditions: • SA z/OS has issued, or will soon issue, the commands. • When SA z/OS attempted to start the application it found that the application monitor status for the application was STARTING. • During initial status determination, SA z/OS found that the application monitor status for the application was STARTING. • SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED, but the application monitor status for the application is STARTING. Note that the relevant automation flag, Initstart or Restart, must be on. The application startup commands as defined in the automation control file are issued after the application is placed in STARTING status. An application remains in STARTING status until either its ACTIVE or its UP message arrives. After the application start delay time, an SA z/OS monitoring check is issued for it. If it is not found, and if the critical threshold is not reached, SA z/OS will attempt to restart the application. If it is found, but is not yet UP, it is put into ACTIVE status.
STARTED2	The application has become active, but has not indicated that it is ready to accept work within its *start delay* and number of start cycles. An application can be put into STARTED2 status in response to the following conditions: • The startup checker found that the application monitor status was still STARTING. • The startup checker was called for an application whose automation status was ACTIVE. • SA z/OS found the Automatic Restart Manager status for the application to be AVAILABLE-TO. • SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is one of AVAILABLE-TO, RESTARTING or RECOVERING. • SA z/OS checked an attempt by Automatic Restart Manager to restart the application and could not find a better status to put it into. Its application monitor status is ACTIVE. This covers situations where the restart checker is unable to determine the Automatic Restart Manager status for the application, or its status is FAILED. An application remains in STARTED2 status until either its UP message arrives or an operator uses the SETSTATE command to change the application status to UP or RUNNING.
STOPPED	The application has been shut down by an external agent, such as an operator cancel. SA z/OS is not permitted to restart it and will not allow the Automatic Restart Manager to restart it. This status is preserved across a recycle of SA z/OS or a re-IPL of the processor unless the application has its Restart after IPL Option set to START or NOSTART. An application remains in STOPPED status until an operator uses the SETSTATE command to change its status to either RESTART or CTLDOWN. An application may also leave STOPPED status if it is restarted outside the control of SA z/OS. In this case, it goes to either the ACTIVE or UP status, depending on which message is received.

Table 8. Automation Agent Statuses (continued)

Value	Meaning
STOPPING	The application is being shut down by an external agent. This status is entered if a TERMINATION message is received but SA z/OS is not in the process of shutting the application down. It may also indicate that the application is abending, but the abend messages issued are being treated as normal termination messages by automation. An application that is STOPPING remains in that status until either an abend message is received that puts it into ABENDING or BREAKING status, or the final termination message for the application is received, at which point the application RESTART option is checked. If RESTART is ALWAYS, the application is put into RESTART status and SA z/OS attempts to restart it. Otherwise, the application goes to STOPPED status. If it is slow to clear the system after the final message is received, it may be placed into ZOMBIE status. If an ACTIVE or UP message is received for the application, its automation status is changed to either ACTIVE or UP, as appropriate. Automatic Restart Manager interaction with this status depends on a number of factors. If the application is deregistered then there is no interaction. If the application is registered then Automatic Restart Manager will attempt to restart it. If the application goes to STOPPED status before the SA z/OS or Automatic Restart Manager exit is invoked then SA z/OS will tell Automatic Restart Manager not to restart the application. If it does not, SA z/OS will tell Automatic Restart Manager that it can restart the application.
STUCK	An application can get STUCK when it is being shut down. This is because SA z/OS has run out of NORM, IMMED, or FORCE shutdown commands.
UP	The application has finished initializing and is ready for work. An application can be put into UP status in response to the following conditions: • SA z/OS has received an UP message for the application. • An operator has used the SETSTATE command to change the application automation status. In this case, SA z/OS assumes that the operator has ensured that the application is actually UP. • During initial status determination the application monitor status was found to be ACTIVE. • SA z/OS found the Automatic Restart Manager status for the application to be AVAILABLE. There are a number of ways for an application to leave the UP status, if: • It is shut down with the INGREQ REQ=STOP command, it goes to AUTOTERM status • It is shut down outside SA z/OS, it goes to STOPPING status • It abends, it might go to STOPPING, ABENDING or BREAKING status • It has problems, it may go to HALTED status • The regular monitor cannot find it, it will call the TERMMSG generic routine • The application abends, SA z/OS does not pick up the abend messages, and Automatic Restart Manager detects that the address space has ended, the application may go to EXTSTART.

Table 8. Automation Agent Statuses (continued)	
Value	**Meaning**
ZOMBIE	When an application is leaving the system it can enter a ZOMBIE status. This indicates that the final termination message for the application has been received but that SA z/OS monitoring still finds the application. SA z/OS retries monitoring after a delay and the application is put into ZOMBIE status if this situation persists for more than eight retries. There are three ways that an application can enter a ZOMBIE status: • If MVS is slow in clearing the application and the termination delay time is short. • If there are two jobs with the same name in the system, one of which is the application. When either of them terminates and SA z/OS does not know the address space ID of the application or does not get the address space ID with the termination message, SA z/OS assumes that the application has stopped, but SA z/OS monitoring will find the other job. To change the status to UP, either manually shut down the other job, or use the SETSTATE command to change the application status back to UP. • The job may have become stuck in the system after issuing its final message. From ZOMBIE status, if the application suffers an unrecoverable abend it will go into BREAKING status. **Note:** The Restart after IPL option of the customization dialog may override these resource statuses at SA z/OS IPL or recycle, resulting in SA z/OS starting the subsystem.

Figure 18 on page 77 and Figure 19 on page 78 indicate the relationships between the automation statuses. You can change the states illustrated here with the SETSTATE command (see “Changing the Automation Agent Status” on page 142).

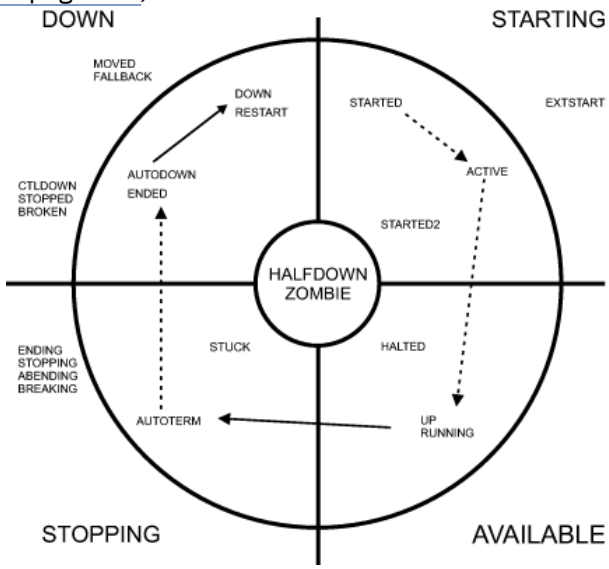


Figure 18. Transitions for a Normal Subsystem

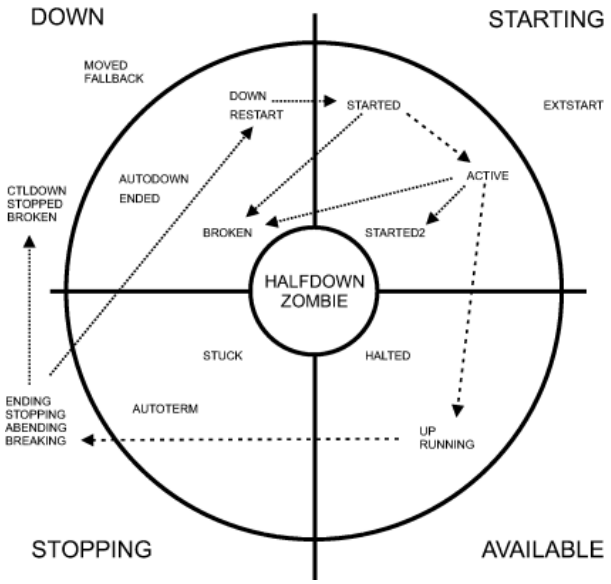


Figure 19. Transitions for an Abending Subsystem

Application Monitor Status

SA z/OS determines the application monitor status of an application by running a routine that you specified in the customization dialog.

You can specify the routine for an individual application, and you can specify a default application monitor routine for applications on an entire system. For further information on application monitor routines, refer to *IBM Z System Automation Defining Automation Policy*.

Table 9. Regular Monitoring: Automation Status Determination

Application Monitor Status	**Automation Status Change**
ACTIVE	The application automation status is changed to ACTIVE.
STARTING	The application automation status is changed to STARTED and a startup checker is scheduled for it.
INACTIVE	If the application is defined with a RESTART option of ALWAYS, the automation status is changed to RESTART and the restart process is invoked. If its RESTART Option is not ALWAYS, its automation status is changed to INACTIVE.

Automatic Restart Manager Statuses

The z/OS Automatic Restart Manager statuses are determined by SA z/OS and are used to help determine automation statuses.

These statuses are included here as background information so that you can better understand the SA z/OS automation statuses. They are a superset of the statuses that Automatic Restart Manager uses, the added statuses being ELSEWHERE and UNREGISTERED. Possible values for the Automatic Restart Manager status are:

Value	**Meaning**
AVAILABLE	The element name is registered with Automatic Restart Manager, the job is running on this system and the Automatic Restart Manager status for it is AVAILABLE.

Value	Meaning
AVAILABLE-TO	The element name is registered with Automatic Restart Manager and the job is running on this system. The job has been restarted by Automatic Restart Manager, has issued its IXCARM REGISTER call, but did not issue its IXCARM READY call within the **ready timeout** defined to z/OS for it.
STARTING	The element name is registered to Automatic Restart Manager and the job is running on this system. The job was started by an agent external to MVS, has issued its IXCARM REGISTER call, but has not yet issued its IXCARM READY call. Unlike the RECOVERING status, this status does not time out to the AVAILABLE-TO status.
RECOVERING	The element name is registered to Automatic Restart Manager and the job is running on this system. Automatic Restart Manager has issued the restart command for the application, it has issued its IXCARM REGISTER call, but has not yet issued its IXCARM READY call. It if takes too long to do this, it is placed in the AVAILABLE-TO status.
FAILED	The element name is registered with Automatic Restart Manager, the job has failed, but was last active on this system. It is unknown at this point if Automatic Restart Manager will attempt to restart the job. SA z/OS will continue to report an elements status as FAILED until MVS either deregisters it, or is committed to restarting it. MVS query commands issued during this time will probably return a RESTARTING status, however MVS Automatic Restart Manager is still consulting its policies and exits to determine whether it should restart the element.
RESTARTING	The element name is registered with Automatic Restart Manager, the job has failed and Automatic Restart Manager is attempting to restart it. This status is entered very early in the restart process, before Automatic Restart Manager is committed to restarting the element. SA z/OS continues to report the status as FAILED until Automatic Restart Manager *is* committed to issuing the restart commands. This means that MVS query commands may report a status of RESTARTING although SA z/OS reports a status of FAILED. The RESTARTING status is exited when the job issues its IXCARM REGISTER call. If it takes too long to do this, its element name is deregistered.
ELSEWHERE	The element name is registered with Automatic Restart Manager, but is currently active on a different system. Unless that system fails, Automatic Restart Manager will not move the application to this system.
UNREGISTERED	The element name is not, currently, registered with Automatic Restart Manager. This may mean that the application has never been active within the sysplex, or it may mean that it failed and Automatic Restart Manager could not recover it. If the SA z/OS routine to determine an Automatic Restart Manager status finds that Automatic Restart Manager recovery is inactive, the UNREGISTERED Automatic Restart Manager status is returned for all enquiries. This means that SA z/OS effectively ignores Automatic Restart Manager.

Mapping the Automation Agent Statuses to the Automation Manager Observed Status

Table 10 on page 80 shows how the current resource statuses that are retrieved by the related automation agents are mapped to the observed status that is used by the automation manager.

Table 10. Automation Agent to Automation Manager Status Mapping

Automation Agent	Observed	Automation	Notes
DOWN	SoftDown	Idle	The automation agent is ready to perform an initial start of the resource.
RESTART	SoftDown	Idle	The automation agent is ready to perform a startup of the resource.
STARTED	Starting	Busy	The automation agent has issued the start commands for the resource.
EXTSTART	Starting	Busy	This means that the automation agent has detected that an external agency (probably ARM) is in the process of starting the resource.
ACTIVE	Starting	Busy	The automation agent has received some indication that the start is successful, although it is not yet complete.
STARTED2	Problem	Problem	Although the resource is still active, it does not appear to have started properly within the time period that the automation agent expected it to start within.
UP	Available	Idle	The resource has entered a stable state and is available for usage.
HALTED	Degraded	Busy	The resource has encountered a problem, but is still active. The automation agent may be attempting to return it to an UP state.
RUNNING	Starting	Busy	The resource is a transient subsystem that is currently executing.
ENDING	Starting	Busy	The resource is a transient subsystems that is terminating.
AUTOTERM	Stopping	Busy	The automation agent is in the process of stopping the resource.
STOPPING	Stopping	Busy	The automation agent has detected that the application is terminating normally outside of its control.
ABENDING	Stopping	Busy	The automation agent has determined that the application is terminating abnormally but can be recovered.
BREAKING	Stopping	Busy	The automation agent has determined that the resource is terminating abnormally in a nonrecoverable fashion.
ENDED	Available	Idle	The resource is a transient subsystem that has completed successfully.
AUTODOWN	SoftDown	Idle	The resource is down and may be restarted as needed by automation.
CTLDOWN	HardDown	Idle	The resource is down and may not be restarted by automation.

Table 10. Automation Agent to Automation Manager Status Mapping (continued)

Automation Agent	Observed	Automation	Notes
STOPPED	HardDown	Idle	The automation agent has determined that the resource has been stopped normally outside the control of automation and its policy indicates that an automated restart is not allowed.
BROKEN	HardDown	Idle	The automation agent has determined that the application has failed in a nonrecoverable fashion.
HALFDOWN	Degraded	Problem	A shutdown for the resource by the automation agent was cancelled at the users request. Although the resource is still active, its status is indeterminate.
STUCK	Stopping	Problem	While trying to stop or start the resource, the automation agent ran out of instructions.
ZOMBIE	Problem	Problem	The automation agent believes that the resource should be inactive, but it is not.
MOVED	Standby	Idle	The automation agent has determined that the resource is active elsewhere within the sysplex. The manager may reactivate it on this system.
FALLBACK	Standby	Idle	The resource is a backup for a resource that normally runs on a different system within the sysplex. The manager may start it if it needs to.

Monitor Statuses Supplied by Automation Agents

Table 11 on page 81 describes the statuses available for a monitor and supplied by the automation agents. The statuses for a monitor include both the monitor status itself and the health status of the monitored object.

Table 11. Monitor Statuses

Status	Description
INACTIVE	The monitor is not running.
FAILED	The monitor has failed. Recovery may be in progress. No acceptable health status was provided.
BROKEN	Both the monitor and recovery failed. This is a permanent condition. The monitor will not be re-invoked.
UNKNOWN	The health status is not available yet.
NORMAL	The monitor has obtained good results from the object, or objects, that is watching.
WARNING	The monitor detected a certain degree of degradation in the operation of the monitored object.
MINOR	The same as WARNING, but more severe.
CRITICAL	The same as MINOR, but more severe.

Table 11. Monitor Statuses (continued)	
Status	**Description**
FATAL	The same as CRITICAL, but more severe.

The significance of a monitoring status, in combination with other application and agent statuses is described in Table 12 on page 82.

Table 12. Monitor Status Mapping

Monitor	Health	Observed	Agent	Meaning
INACTIVE	UNKNOWN	UNAVAILABLE	IDLE	The monitor is not running.
STARTING	UNKNOWN	STARTING	BUSY	The monitor is about to start. An activate command might be in progress.
ACTIVE	UNKNOWN	AVAILABLE	IDLE	The monitor is running. No health status has been determined yet. This might be a passive monitor.
ACTIVE	NORMAL	AVAILABLE	IDLE	The monitor is running and good health is reported.
ACTIVE	WARNING	AVAILABLE	IDLE	The monitor is running and a problem with health has been reported.
ACTIVE	MINOR	AVAILABLE	IDLE	Same as ACTIVE/ WARNING but more severe.
ACTIVE	CRITICAL	AVAILABLE	IDLE	Same as ACTIVE/ MINOR but more severe.
ACTIVE	FATAL	AVAILABLE	IDLE	Same as ACTIVE/ CRITICAL but more severe.
FAILED	UNKNOWN	AVAILABLE	IDLE	The monitor is running. A health status could not be determined.

Table 12. Monitor Status Mapping (continued)

Monitor	Health	Observed	Agent	Meaning
STOPPING	UNKNOWN	STOPPING	BUSY	The monitor is about to stop. A deactivate command might be in progress. The health status is maintained from the last ACTIVE status.
BROKEN	UNKNOWN	HARDDOWN	IDLE	The monitor suffered a nonrecoverable problem and is therefore stopped. SA z/OS will not restart it.

SA z/OS Processes Involved in Status Determination

This section describes a number of SA z/OS processes that are involved in determining the various statuses of resources. A resource's status is determined by the automation agent and reported to the automation manager.

Initial Status Determination
This process occurs at the end of SA z/OS initialization and after a configuration refresh. It attempts to determine a suitable automation status for each application that is defined to SA z/OS.

Subsystem Startup
This process is invoked whenever SA z/OS is about to start an application. It checks that conditions are right for the application to be started and that SA z/OS is allowed to start the application. If appropriate, the subsystem startup process then issues the commands to start the application and changes the application automation status to STARTED.

Startup Checker
This process is run for each application after SA z/OS has issued the startup commands, after the amount of time specified by the Start Timeout defined for the application. It changes the automation status from STARTED to STARTED2 if the application fails to start within its expected start delay.

Regular Monitoring
This process runs at an interval defined by the automation administrator in the customization dialog. Normally the time interval is between 30 minutes and 12 hours. The purpose of regular monitoring is to carry out a reality check to ensure that the current SA z/OS automation status for an application is consistent with the current application status as determined by the application monitor routine.

To help determine consistency, SA z/OS statuses are divided into four groups:

UP STATUSES
These statuses mean that the application monitor status should be ACTIVE. They include:

- UP
- RUNNING
- ACTIVE
- HALTED

DOWN STATUSES
These statuses indicate that the application monitor status should be INACTIVE. They include:

- DOWN
- RESTART
- INACTIVE
- AUTODOWN
- CTLDOWN
- BROKEN
- STOPPED
- ENDED
- FALLBACK
- MOVED

STARTING STATUSES
These statuses indicate that the application monitor status may be STARTING or ACTIVE. They include:

- STARTED
- STARTED2
- EXTSTART

TRANSITIONAL STATUSES
These statuses are either part of the shutdown process, or indicate that the application may or may not be present in the machine. *All* application monitor statuses are possible with any transitional status. They include:

- AUTOTERM
- STUCK
- ZOMBIE
- BREAKING
- ABENDING
- HALFDOWN
- STOPPING
- ENDING

If regular monitoring finds a status mismatch, it changes the application automation status according to Table 9 on page 78.

It should be noted that several other routines have similar status mismatch detection capabilities built in.

Secondary Monitor
This is invoked, after a small delay, from the application monitor routine. This process performs a recheck for a specific application that the application monitor routine found to be in an unusual (generally a transient) state. It is responsible for making some of the transitions outlined under the Routine Monitor process.

Shutdown
This process is normally invoked by an operator to shut down one or more applications. The shutdown process is responsible for the sequencing of the various shutdown commands. Depending on the parameters it is invoked with, the application that is shut down may be put into an AUTODOWN or CTLDOWN automation status, or it may be restarted.

Resynchronization Process (Environmental Analysis Process)
It may be invoked manually by an operator using the RESYNC command. This process assesses the current status of the various parts of the z/OS system that SA z/OS automates. It issues display commands to obtain information on WTORs and sysplex resources. It also invokes the Initial Status Determination process. When it runs, all previous status information is discarded.

Suspend and Resume Functionality

Normally, SA z/OS automates resources that are defined in the policy based on their desired status goals (that is, to be available or unavailable). If the goals are satisfied, no action is taken. But SA z/OS reacts each time an automated message arrives or whenever the actual observed status differs from a resource's desired status. Resources can be started or stopped; commands can be issued as result of messages; operators can be alerted, and so on.

In some situations, for instance, when maintenance activities require manual startup or shutdown of a resource, the involvement of SA z/OS is not wanted. In fact, it would be even counterproductive if SA z/OS "corrected" that.

In such situations, SA z/OS allows the operator or administrator to suspend a resource. While it is suspended, automation does not attempt to start or stop this resource. Similarly, it does not react on messages that would normally trigger status commands or other commands that are defined in the policy for a message. Most importantly, it does not alert the operations team by exposing an unusual Automation Agent or observed status on any status display (for instance INGLIST or SDF), which operators are normally sensitive to.

Suspending a Resource

A resource is suspended using the INGSUSPD command. This command accepts resource types APL, APG, MTR, and REF. Like requests to start or stop the resource, INGSUSPD produces a request called suspend request.

Using INGVOTE, suspend requests can be viewed, canceled, and killed. The latter two actions refer to resuming the resource.

A suspend request is handled by the automation manager but processed asynchronously by the automation agent. It succeeds, if there is no startup or shutdown currently in progress for a given resource. If there is a startup or shutdown in progress, the suspend request is still recognized but is pending; it will be honored only once the desired status is reached.

Similar to INGREQ for start or stop requests, SA z/OS allows you to specify a SCOPE parameter of the INGSUSPD command to suspend just a single resource or all resources in the dependency chain as illustrated in Figure 20 on page 85. For a SCOPE=ALL request, suspend votes are propagated along the dependency chain to all dependents.

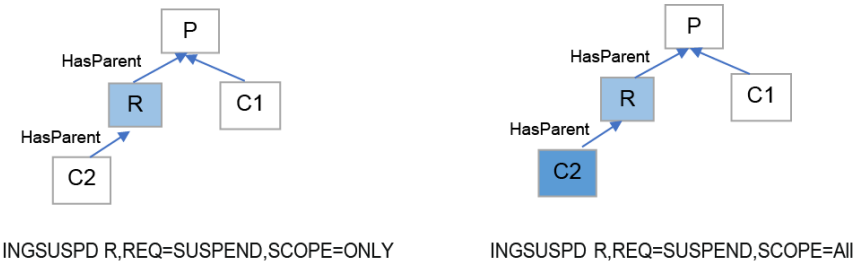


Figure 20. Scope Parameter of INGSUSPD Command

The SCOPE parameter can be a powerful way to suspend a whole tree of resources at once. For the effect of this parameter when used for groups, see also “Suspending APGs or Its Members” on page 87.

Reporting Suspended Resources

For all supported resource types, SA z/OS exposes the automation manager suspend flag value among others on the INGLIST panel and by the INGDATA command as shown in Table 13 on page 86. For MTR types, the status is also shown on the DISPMTR panel.

Table 13. Suspend Flag	
Status	**Description**
blank	The resource is not suspended.
DIR(ect)	The resource is suspended using INGSUSPD or the suspend file.
IND(irect)	The resource is suspended because a supporting resource is suspended using INGSUSPD or the suspend file.
PEN(ding)	The suspend request is recognized but the resource is currently being started or shut down. Therefore, the request is held back until the desired status is reached.

To programmatically determine whether a resource is suspended or not, use INGDATA. If you want to simply test whether an APL can be automated on the automation agent side, call AOCQRY and check return code 0. A return code 1 means that the Automation flag was turned off or that the resource is suspended.

To differentiate between the turned off automation agent flag or a suspended APL, you can refer to the task global variables *SUB*SUSPEND* provided with AOCQRY.

While a resource is suspended, SA z/OS still monitors whether it is active or inactive. However, unlike non-suspended resources, the lifecycle status model for suspended resources merely distinguishes two status values for any given status type as shown in Table 14 on page 86:

Table 14. Lifecycle Status Model for Suspended Resources			
Type	**Automation Agent Status**	**Monitor Resource Status**	**Observed Status**
APL	UP	x	AVAILABLE
	DOWN	x	SOFTDOWN
Transient APL	ENDED	x	AVAILABLE
	RUNNING	x	STARTING
APG	x	x	AVAILABLE
	x	x	SOFTDOWN
MTR	x	ACTIVE	AVAILABLE
	x	INACTIVE	SOFTDOWN

When the observed status differs from the desired status at the time when you resume a resource, SA z/OS immediately attempts to bring it to its desired status as usual.

Overruling a Suspended Resource

The most prominent use case, when overruling a suspended resource becomes necessary, is when you want to perform a system shutdown. In this situation, you do not want the shutdown process to be delayed just because the suspended resource cannot be shut down.

To enable SA z/OS to shut down such a suspended resource, it provides a new override flag, called SUS, which you can specify with the INGREQ and INGRUN commands. The mechanism works for all start and shutdown requests, not just the system shutdown. For a system shutdown, when INGREQ *ALL is specified, or when SYG is in the scope of a stop request (for example using wildcards or selecting SYG directly), the override flag is provided automatically.

The effect of using OVERRIDE=SUS is that SA z/OS sends an order regardless of the value of the suspend flag to bring the resource into its desired status. It means that the conditions to start or stop other resources can become satisfied and hence ultimately can successfully complete the original request. Note that on the automation agent side, the OVERRIDE=SUS option is handled as if OVERRIDE=FLG is given, that is, commands defined in the policy are issued, even though the resource is suspended.

Suspending APGs or Its Members

SA z/OS also supports suspending Application Groups (APGs) and their members. But be aware that suspending only a member, leaving the containing group unsuspended, might lead to group actions for APGs of nature MOVE and SERVER, depending on what happens to this suspended member.

Taking Figure 21 on page 87 for example, assume that member M1 is currently selected in the MOVE APG G and that this member is active. If it is suspended using INGSUSPD, nothing happens - M1 remains active. However, as soon as you manually stop M1, the automation manager will look for another viable member in G, find M2, and start it.

Continuing the scenario above, assume that M1 (inactive, suspended) has a base preference value of 951 and M2 (active and selected) has an adjusted preference value of 950 (base preference of 700 plus availability bonus of 225 plus sticky bonus of 25). As soon as M1 is resumed, SA z/OS performs a disruptive move by shutting down M2 and starting M1 because M1's preference value is higher than M2's.

To avoid such frequent unexpected group actions, it is recommended to suspend the whole group. Using the SCOPE parameter, you can decide whether only the group and its members or in addition all the descendants of the group and the descendants of the individual group members are suspended, too. The following figure illustrates the difference:

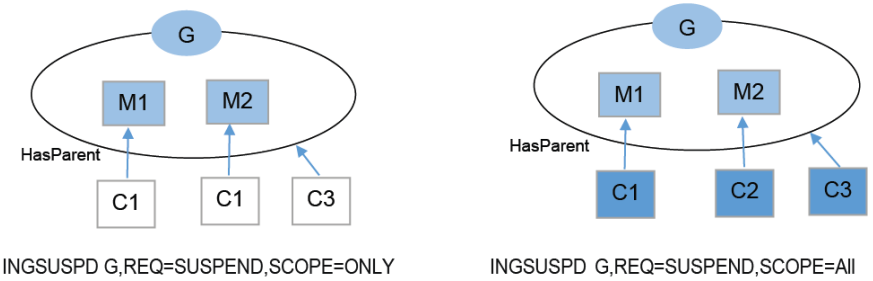


Figure 21. Suspend a Group and Its Members Using Scope Parameter

The parameter SCOPE=ONLY suspends the group and its members, but no dependant resources; neither dependants of the group itself (like C3) nor dependants of its members (C1 and C2). In contrast, the parameter SCOPE=ALL includes the suspension of all descendants of the group, direct and indirect descendants.

If a member is suspended directly or indirectly through the suspended group, any start or stop votes previously propagated from the group to the member and further down to all its dependants are withdrawn. For suspended members of a group of nature SERVICE or MOVE, the desired status will therefore change to the default desired status, which is usually AVAILABLE, as the member will not have a group vote propagated to it. This may result in a compound status of INHIBITED for those dependant resources that are currently not suspended. Consider using SCOPE=ALL when suspending the group in this case to avoid INHIBITED.

Taking Figure 21 on page 87 for example, assume that member M2 is currently not selected in the MOVE APG G and this member is inactive. If it is suspended using INGSUSPD, the desired status for M2 will be changed to AVAILABLE, but automation will not trigger any action.

However, as soon as you resume the member using INGSUSPD, the automation manager may generate a stop vote again, changing the desired status to UNAVAILABLE if another member is still selected by the group based on current preference values.

Suspending Monitor Resources

If an MTR is suspended, no monitoring (neither active, nor passive) takes place anymore. The health status is set to UNKNOWN and the health status of all resources that have defined a HasMonitor relationship to this MTR is updated accordingly.

Once the MTR is resumed, the monitor command, if applicable, will be invoked again at the intervals specified in the policy and the MTR will honor messages again that can result in a new health status.

Note that the MTR's activate and deactivate commands are not invoked when you suspend or resume, respectively, an active MTR.

Customizing and Programming Considerations

SA z/OS also allows you to inquire the value of the automation manager suspend flag from within a message trap using the automation table function ING$QRY.

INGSUSPD provides the same level of security compared to INGREQ. For more information, see INGSUSPD command in *IBM Z System Automation Operator's Commands* and "Security and Authorizations" in *IBM Z System Automation Planning and Installation*.

Exit routine AOFEXC26 is invoked by INGSUSPD, if applicable, to determine whether to proceed with the command or not. If all you want is to enforce that the user provides a comment along with the request, copy the sample out of SINGSAMP into your DSICLD dataset concatenation. For more information, see "Command Exits" in *IBM Z System Automation Customizing and Programming*.

Other considerations

1. An MTR can always be suspended by itself. However, in most cases, you want to suspend monitoring automatically at the same time when the monitored APL or APG is suspended. So, to ensure that you do not have to perform this extra step, make sure that you also specify a HasParent relationship from the MTR to the related APL or APG and use the SCOPE=ALL parameter to propagate the request to the MTR.
2. If an APL is suspended while Job Log Monitoring is active, that function is turned off for this resource. Also, the function is not turned on automatically upon resuming the resource. To enable Job Log Monitoring again, you must do it manually using INGJLM, or make sure that the resource is first resumed and a start request or vote exists for SA z/OS to start it.
3. While suspended, with one exception, setting any status or flag using INGSET or SETSTATE is prohibited. Note: Both commands should be used with care, anyway, and only by users who understand their implications as they might confuse the internal status bookkeeping of SA z/OS.
4. On the automation agent side, when a resource is suspended, all automation flags (Automation, Initstart, Start, Recovery, Terminate, and Restart) are set to 'S' and that setting is also propagated down to minor resources, if applicable. While being suspended, DISPFLGS and DISPSTAT will not allow you to change the status of any of these automation flags. You must first resume the resource before you can change the status of a flag.
5. It is not recommended to use the INGSUSPD for resources running on an automation agent that has been suspended by the INGAMS command. Otherwise, it may cause mismatches of a resource's suspended status between the automation manager and the automation agent.
6. When you also use the Application Restart Manager (ARM) to manage resources, note that SA z/OS will tell ARM to not perform an action on behalf of a suspended resource. Also note that the use of ARM is no longer necessary these days as SA z/OS offers sysplex-wide MOVE groups.
7. Note that the following capabilities stay in place, even while a resource is suspended:
   - OUTREP processing takes place to save the reply ID, if a WTOR was received that is normally handled for the resource.
   - Exit routines registered at the Status Observer by a user script or registered internally by SA z/OS are called when the resource changes.

- The status of a Workload Scheduler (ZWS) special resource follows any status change of the resource.
- Message capturing, threshold handling, and WLM resources management are done by default. You can control these behaviors in the Customization Dialog > System Defaults > AUTOMATION OPTIONS policy. For more details, see "System Defaults Entry Type" in *IBM Z System Automation Defining Automation Policy*.
- The resource is not exempted from IPL-complete monitoring, which means that posting completion is likely to be delayed unless such a resource is started manually.

8. Refer to *IBM Z System Automation Customizing and Programming* for further details on the commands mentioned in this section and to *IBM Z System Automation Planning and Installation* for details on the services you can use in your own commands and scripts.

Using the Suspend File

Resources that are already automated can be suspended and resumed during runtime using the INGSUSPD command. But there might be situations, where it is useful or necessary to "plan" resources in the policy, which are not ready to be used.

For instance, an SA administrator adds an application to the policy and the application is not yet installed. If a policy with such "planned' resources is loaded, they are automated (started and stopped) and may produce error situations until somebody suspends them manually with the INGSUSPD command. To prevent such error situations, it is possible to leverage the planned suspend capabilities of so called *suspend file*.

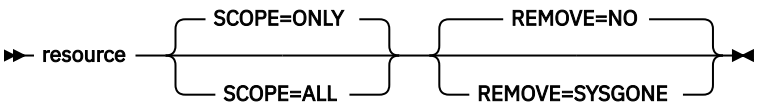
This suspend file is processed during a manager COLD/WARM start or an automation configuration REFRESH. A suspend request is generated for all resources listed in the suspend file, so they are not automated after the load of a new or changed policy.

The Suspend File

Properties of the suspend file

- Sequential data set or member of partitioned data set
- The suspend file name is specified in the SUSPENDFILE parameter within the PARMLIB member HSAPRM*xx* of the automation manager. See "Configuring HSAPRMxx" in *IBM Z System Automation Planning and Installation*.
- Format: no restrictions; example FB 80, VB 255

Syntax of the suspend file



resource

Specifies the name of the resource to be processed. The format is name[/type[/system]]. Only resources of type APL, APG, MTR, and REF are supported. This is a mandatory parameter.

Note: This parameter must not start at column 1. If you start your definitions in column 1, the first character will be overwritten during the processing of the suspend file (see “Processing the Suspend File” on page 90).

SCOPE

Specifies whether the SUSPEND affects only the resource or both the resource and its descendants. This is an optional parameter. Valid options are:

ONLY

Specifies that only the specified resource is suspended. This is the default.

ALL

Specifies that the resource and its descendants are suspended.

REMOVE

Indicates the condition when the request is automatically removed, regardless of whether the request is satisfied or not. If the specified condition matches the observed status of the resource, the request is deleted. This is an optional parameter.

SYSGONE

If the system, where the automation agent runs, leaves the sysplex, the observed statuses of all resources running on that system become SYSGONE.

NO

The suspend request is not automatically removed, regardless of the observed status of the resource. This is the default.

You can add descriptive information in the file with `!` or `x'5A'` depending on your code page at the beginning of each line, which indicates that the rest of the line should be treated as a comment. Blank lines are ignored.

Processing the Suspend File

The suspend file is processed during a manager COLD/WARM start or a configuration REFRESH.

The automation manager requires exclusive access to the suspend file while REFRESH or COLD/WARM start. If anybody is currently reading or writing the suspend file, the processing of the file will fail with HSAL1118I message.

Manager REFRESH

Before you begin

To process the suspend file with a configuration REFRESH, it is necessary that the configuration file has changed. Only changing the suspend file will not cause its processing.

Procedure

The processing of the suspend file during a REFRESH consists of the following steps:

1. **Preprocessing**: The file is read and checked for errors (for example, syntax errors, I/O errors)
2. **No errors occurred during preprocessing:**
   a) Suspend requests are injected for all resources in the suspend file with the parameters defined in the file as well.
   b) All lines that have been processed are commented out using the line comment character in column 1.

   Note: If column 1 is not empty, the first character will be overwritten with the line comment character.
3. **Errors occurred during "preprocessing":**
   a) Error messages are written to the system log.
   b) Config REFRESH is canceled and can be retried later.
4. **Resources defined in the suspend file are not part of the new or changed configuration:**
   a) Informational messages are written to the system log.
   b) Non-existing resources are ignored; all lines with existing resources are processed as described in step 2.

5. **If you want to force the REFRESH:** use the **INGAMS** command (see INGAMS) or the **MVS Modify** command (see “Starting, Stopping, and Maintaining the Automation Manager” on page 101).

   For this purpose, a new parameter for both commands is introduced, which allows to specify the following:

   - NOSUSPEND: Ignore the whole suspend file
   - FORCESUSPEND: Ignore all erroneous lines in the suspend file

   Note:

   - The **INGAMS REFRESH** or **MVS Modify** command with FORCESUSPEND option or without any option will process the suspend file only if the configuration file has been changed as well. For example, if you have added, deleted, or modified a resources.
   - This functionality can also be used, if there are no errors in or with the suspend file. For example, you want to REFRESH the configuration and completely ignore the suspend file.

Manager COLD/WARM start

Procedure

The processing of the suspend file during a manager COLD/WARM start consists of the following steps:

1. **Preprocessing**: The file is read and checked for errors (for example, syntax errors, I/O errors).
2. **No errors occurred during preprocessing**:
   a) Suspend requests are injected for all resources in the suspend file with the parameters defined in the file as well.
   b) All lines that have been processed are commented out using the line comment character in column 1.

      Note: If column 1 is not empty, the first character will be overwritten with the line comment character.
3. **Errors occurred during preprocessing**:
   a) Error messages are written to the system log.
   b) HSAM1318D WTOR is issued with the following options. The automation manager initialization waits until the operator enters the corresponding input data.

      NOSUSPEND
      Continues the automation manager initialization without processing the suspend file. No suspend requests are processed.

      FORCESUSPEND
      Continues processing the suspend file and ignores the errors (only for syntax errors).

      CANCEL
      Stops the automation manager initialization at all.

      If the input data is invalid, the HSAM1319I message will be issued and HSAM1318D asks again for input.

      If the input is valid, the automation manager initialization continues using the specified suspend file option.
4. **Resources defined in the suspend file are not part of the new or changed configuration**:
   a) Informational messages are written to the system log.
   b) Non-existing resources are ignored; all lines with existing resources are processed as described in step 2.

Dynamic Resources

The concept of dynamic resources was introduced to primarily address automated (DevOps) deployment scenarios for single address spaces, for example, APLs representing CICS and IMS regions and USS resources. Its purpose is to have one or more predefined templates in the policy that can be used by operators or automation to dynamically bring these resources into the scope of SA z/OS without running through a policy build and refresh cycle.

The general steps to use dynamic resources and a few things to consider are as follows. This process enables you to apply enterprise policies to the content of a template via the existing change control mechanisms, while giving additional speed and flexibility to create resources at runtime based on template definition.

1. Set up the backend database to store dynamic resources.

   System Automation offers two options to store dynamic resources: its internal data store and IBM Db2 for z/OS (the later option is enabled in OA63123). Technically, dynamic APL resources are persisted in the backend database rather than the policy and automation control file. Therefore, it is a prerequisite to run through the documented setup for the backend database when you plan to use dynamic resources.

   You need to choose between the two database options. You cannot switch back and forth between the database. Once you choose between anyone, the data would be fetched and inserted into the database chosen.

   - If you choose System Automation data store, see "Step 27: Configure and Run the System Automation Data Store" in *IBM Z System Automation Planning and Installation*.
   - If you choose IBM Db2 for z/OS, see "Step 28: Configure Db2 as an alternative database of dynamic resources" in *IBM Z System Automation Planning and Installation*.

   Note: If the database is not available, INGDYN will issue an error. Also, in a policy refresh or automation restart scenario, the creation of dynamic resources depends on the availability of the database. In case of issues with the database availability, the automation of dynamic resources is delayed in such a refresh or restart scenario until the database is available again.
2. Define one or more APLs with type TEMPLATE in the Customization Dialog.

   Note:

   - Templates support only limited CLASS inheritance. If class inheritance is used, relationships, trigger, service periods, runtoken, and desired available status are not inherited. Since templates can be considered as a 'class concept' itself, it is recommended to specify everything on the template.
   - Dynamic resources cannot be supporting resources for other resources.
   - If you link a template to a service period (SVP), ensure that the SVP is already linked to a resource, such as an APL, APG, or MTR. SVP is not generated into the ACF at build time if it is used only with templates. Therefore, if you link a template to an SVP that's not linked to any resource, the dynamic resource that is created based on that template will not include the linked service period.
   - If you link a template to a pacing gate, ensure that the pacing gate is already linked to at least one static APL instance or class. Otherwise, the pacing gate is not generated for the template.
3. Build and refresh the configuration. No automation resources are created at build time. However, the template information is made available to the agents on all systems in the SAplex.
4. Operators can use the INGDYN command at runtime to create dynamic APL resources based on the available templates. No build and refresh is needed.

   Note:

   - A dynamic resource that is created by INGDYN is initially placed into a suspended state. It is expected that an operator or other automation (for example, via INGDYN AOFEXC29 exit) resumes the newly created resource.
   - If a dynamic resource is added to a hosting APG during INGDYN creation processing and this group is a MOVE or SERVER group, the resource will be added as an 'operator' role for model 2 groups and

with preference value 1 for model 1 groups. It is expected that an operator or other automation sets the expected role or preference value via INGGROUP command before resuming it.

5. (Optional) You might need to turn a dynamic resource to a static resource and add its definitions in the policy. When turning a dynamic resource to a static resource, you need to manually delete the dynamic resource.

   Static resources and dynamic resources are both stored in the data model of automation managers and agents. In addition, dynamic resources are also stored in the backend database, either in the System Automation internal datastore or IBM Db2/OS. Automation managers and agents uniquely identify static and dynamic resources by their names. If you add a static resource with the same name as a dynamic resource, the dynamic resource will be overwritten by the static one in the data model of automation managers and agents. Additionally, the record of the dynamic resource is not removed automatically from the datastore; therefore, you have to manually delete it. For this reason, when turning a dynamic resource to static, follow these ordered steps:

   a. Assume that there is a dynamic resource, for example, OI1M0405, previously created with INGDYN.
   b. Create a static resource with the same name OI1M0405 in the policy database.
   c. Build the policy database.
   d. Before refreshing the policy, use the **INGDYN DELETE** command to delete the dynamic resource OI1M0405. This command will delete it from both the data model of automation managers and agents and the datastore.
   e. Issue **INGAMS REFRESH** to load the automation control file containing the static resource OI1M0405.

In summary, dynamic resources enable you to apply enterprise policies to the content of a template via the existing change control mechanisms, while giving additional speed and flexibility to create resources at runtime based on template definition.

Using SA z/OS for Monitoring

As an operator, you can monitor and display information about the resources using the SA z/OS 3270-type interface. This interface provides monitoring capability for:

- Processor operations
- System operations via Status Display Facility (SDF), displaying status information (INGLIST) and detailed resource information (INGINFO and DISPINFO for the automation agent view)

You can also monitor messages and alerts. More information about this topic is given in “How to Monitor Alerts on a NetView Console” on page 153.

If SA z/OS has been configured properly, status mismatches between the actual and the monitored status should not occur. In the customization dialog, the automation administrator can specify a monitoring interval for every resource individually. If such an individual interval is not defined for a resource, every 59 minutes (which is the default, but the interval is user specifiable) SA z/OS will run a monitor routine to check the resources' status as known to SA z/OS versus its actual status. If the result indicates an error, its automation status will be updated.

In the customization dialog, the automation administrator can define either an individual monitor routine per application or a default monitor routine for a system that will monitor all applications running on that system. You can manually invoke an application's related monitor routine through the `MONITOR` command. You can use a wildcard for the applications (subsystems) that you want to be monitored, for example, use *abc* for all subsystems containing the string 'abc' in their names.

Resynchronization

If you believe that a lot of the status information provided by SA z/OS is badly out of sync with reality, you can request that SA z/OS reruns its complete subsystem resynchronization process.

This is achieved with the RESYNC command, but it should be used with care. The resynchronization can take a few minutes and during this time a lot of SA z/OS commands are not available.

Inform List concepts

SA z/OS presents information about automation resources such as applications (APL), application groups (APG), monitor resources (MTR) or MVS components (MVC) on NetView panels shown by the corresponding operator commands.

For example, the commands DISPINFO and INGINFO show details for a particular subsystem.

These panels, however, are not designed for continuous monitoring of the operating environment. Instead, important information such as status changes or abnormal conditions need to be forwarded to consoles or other targets that are better suited for these tasks.

Therefore SA z/OS allows you to register automation resources for one or more targets and in case there are situations where these targets need to be informed, SA z/OS automatically forwards this information to these targets, provided they have been set up and configured accordingly.

The **Inform List** field can be specified for individual resources (APL, APG, MTR), MVS component defaults (MDF), system defaults (SDF), or sysplex defaults (XDF). Furthermore, you can set Inform List for individual message IDs in the CAPMSGS pseudo message in Message/User Data policy item (APG, APL, MTR, MVC).

In the following sections, the supported targets are described in more detail.

Inform SDF

The Status Display Facility (SDF) is informed about any status change of automation resources. For example, if an APL fails, the transitional ABENDING status will be forwarded directly to SDF resulting in a real-time update of the corresponding console.

SDF also receives any exceptional message that has been captured either by means of the policy or directly (implicitly by SA z/OS or explicitly by user-defined Automation Table entries or scripts).

For details about using SDF, refer to “Monitoring with the Status Display Facility” on page 153. For information about customizing SDF, refer to *IBM Z System Automation Planning and Installation* and *IBM Z System Automation Programmer's Reference*.

Inform IOM, EIF, USR, TTT

The targets IOM, EIF, USR and TTT all have in common that they are intended for alert-based notification.

Alerts are generated at an alert point, which can be pre-defined by SA z/OS, user-defined by means of using the INGALERT command or when exceptional messages are captured with a severity of CRITICAL.

The keywords have the following meaning:

IOM
Alerts are sent to System Automation for Integrated Operations Management – SA IOM.

EIF
Alerts in form of so-called *Event Integration Facility* (EIF) events are sent to a software product such as Tivoli Netcool OMNIbus.

USR
Customized alert information can be sent by a user-defined alert handler to some user-defined target.

TTT
Alerts can be transformed to incident records (Tivoli Trouble Tickets) via the Tivoli Directory Integrator (TDI) and then sent to the IBM SmartCloud Control Desk.

For more details about using alert-based notification, refer to *IBM Z System Automation Customizing and Programming* and refer to the INGALERT and INGCNTL commands in *IBM Z System Automation Programmer's Reference*.

Inform SMF

When the **Inform List** field contains **SMF**, status changes for APL, APG, and MTR resources are reflected by an SMF record of type 114. To write SMF records of type 114, SMF recording must be enabled.

For reporting purposes, SMF records have to be dumped into a sequential data set first and then processed by the INGPUSMF batch utility, which creates a tabular report suited for a spreadsheet application.

For more information about using SMF reporting, refer to the chapter "Availability and Recovery Time Reporting" in *IBM Z System Automation Customizing and Programming*.

System Automation Info Broker

To forward SDF messages into Automation Dashboards for Z Automation Web Console, where the operators can easily monitor SDF messages, delete captured messages, or reply to WTOR messages in the modern UI, a new optional component called System Automation Info Broker is introduced in System Automation 4.3.

Overview of System Automation Info Broker

The main idea of the System Automation Info Broker component is to provide an infrastructure that can forward System Automation event messages from *producers* to a central repository that is an Apache Kafka® system, where the messages can be consumed by different *consumers*.

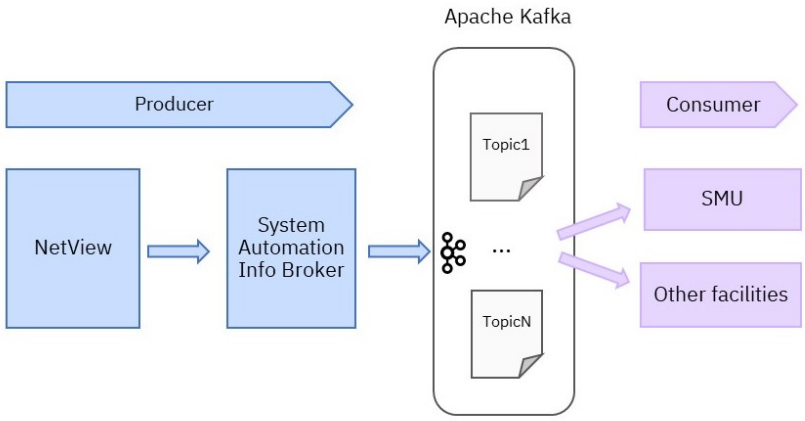


Figure 22. System Automation Info Broker

Apache Kafka is an event-streaming software platform for handling real-time data feeds. It's not delivered with Z System Automation. You must install and start it before you can use the System Automation Info Broker. The producers that are running in NetView sends messages to different Apache topics in a JSON format via the Info Broker. The consumers, for example, Z Automation Web Console and other modern display facilities, can receive and consume incoming messages from Apache Kafka after they subscribed to one or more Kafka topics.

Main components

The System Automation Info Broker contains two main components:

System Automation Info Provider

System Automation Info Provider receives messages from NetView and forwards the messages to the Info Topic Provider (Kafka). It contains two sub-components.

NetView Info Broker Hub

The NetView Info Broker Hub is a Data Services Task (DST) that is started in NetView. The DST gathers, records, and manages message data; and communicates the data with the Info Broker Publisher trough a Unix pipe that is created by the DST.

The NetView Info Broker Hub DST itself also starts a USS address space hosting the Info Broker Publisher.

Info Broker Publisher

The Info Broker Publisher is a Java application that runs in USS and is started by the NetView DST initialization member INGKINIT. This Java process listens to data in the pipe and forwards it to the Info Topic Provider (that is, the Apache Kafka server), by using the Kafka producer API that is available in Java.

Info Topic Provider (Kafka)

The Info Topic Provider is Apache Kafka. Messages are sent into a Kafka topic that's created on the Kafka server before. Consumers (for example, Z Automation Web Console) can subscribe to the topic and consume the messages.

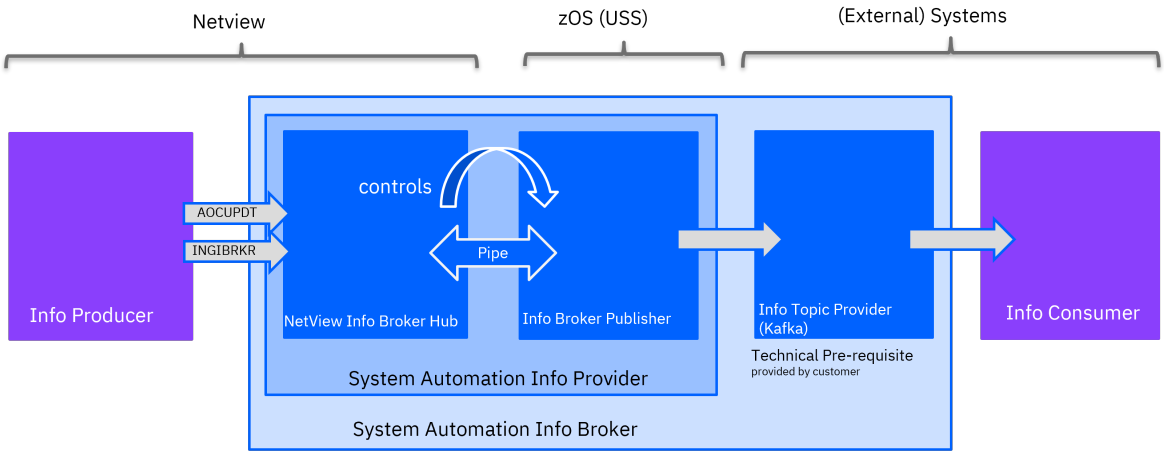


Figure 23. System Automation Info Broker main components

The messages that can be forwarded are SDF messages, and user-defined messages. A new INGIBRKR command was introduced to forward the messages from the producer to NetView Info Broker Hub.

For more information about System Automation Info Broker, you can also watch this video: System Automation Info Broker Introduction.

Configure, Start, and Stop the System Automation Info Broker

For more information about how to configure, start, and stop the System Automation Info Broker, see Configure System Automation Info Broker in *Planning and Installation*.

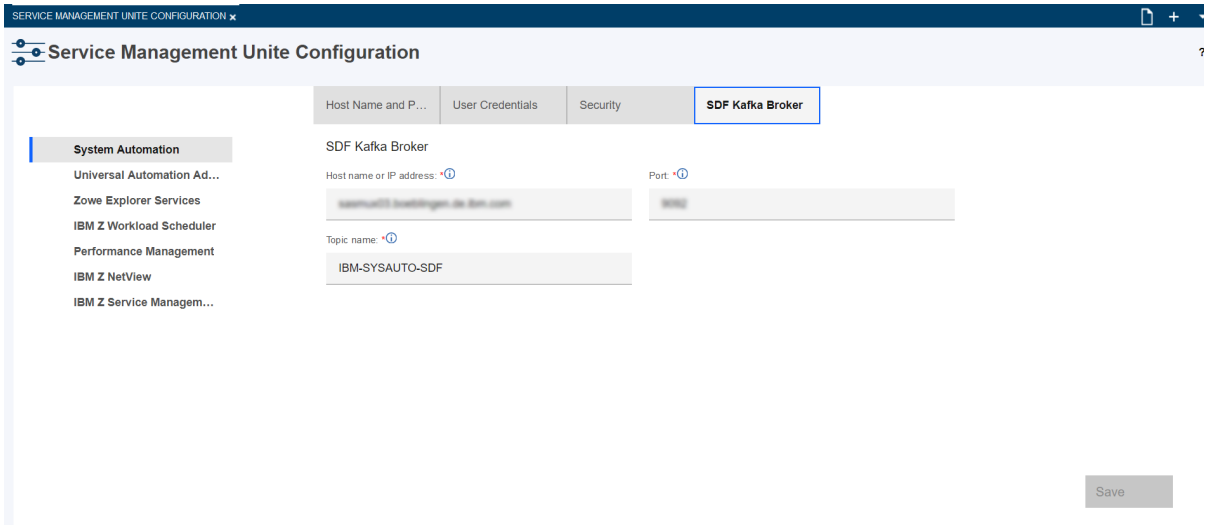
User scenarios of System Automation Info Broker

You can use the System Automation Info Broker to send SDF or user-defined messages to the Apache Kafka topic, where the messages can be consumed by the consumers.

Usage scenario 1: forwarding SDF messages to IBM Z Automation Web Console

The first implementation of this System Automation Info Broker component is to forward SDF messages to Z Automation Web Console via Apache Kafka. Before Z Automation Web Console can receive SDF

messages, you need to configure the following settings in the **System Automation** < **SDF Kafka Broker** tab on the **IBM Z Automation Web Console Configuration** dashboard.



- **Host name or IP address**: The host name or IP address of the Kafka Broker.
- **Port**: The port number of Kafka Broker. The default port is 9092.
- **Topic name**: The topic name of the Kafka Broker to which the System Automation Info Broker sends messages, and from which Z Automation Web Console pulls SDF messages. The value must exactly match the topic name that is configured in the **TOPICSDF** parameter of the System Automation Info Broker initialization member INGKINIT.

After those settings, Z Automation Web Console can receive SDF messages from the subscribed topic. You can view captured messages and their details, and view or reply to WTOR messages on the Z Automation Web Console **SDF Messages Overview** dashboard.

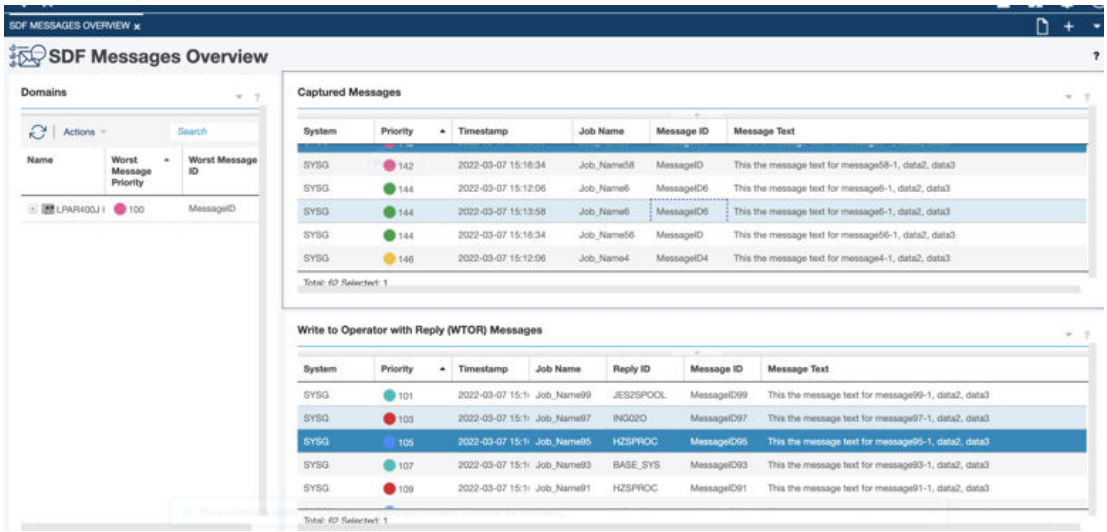


Figure 24. SDF Message Display in Z Automation Web Console

If you want to program your own consumer to consume the SDF messages, see SDF-Message JSON Format in Apache Kafka in *Programmer's Reference* to refer to the format and attribute explanations.

For more information and a demo of this scenario, watch this video: System Automation Info Broker Usage Scenario 1: SDF Messages Forwarding.

Usage scenario 2: forwarding user-defined messages to Kafka consumers

Except for SDF messages, you can also forward any user-defined messages via the System Automation Info Broker component by using the new INGIBRKR command. For usage examples, see INGIBRKR in *Programmer's Reference*.

For more information and a demo of this scenario, watch this video: System Automation Info Broker Usage Scenario 2: User Messages Forwarding.

Troubleshoot System Automation Info Broker

This topic describes the possible error scenarios of the System Automation Info Broker and how you can spot and troubleshoot them.

Data Services Task (DST) issues

Possible error scenarios

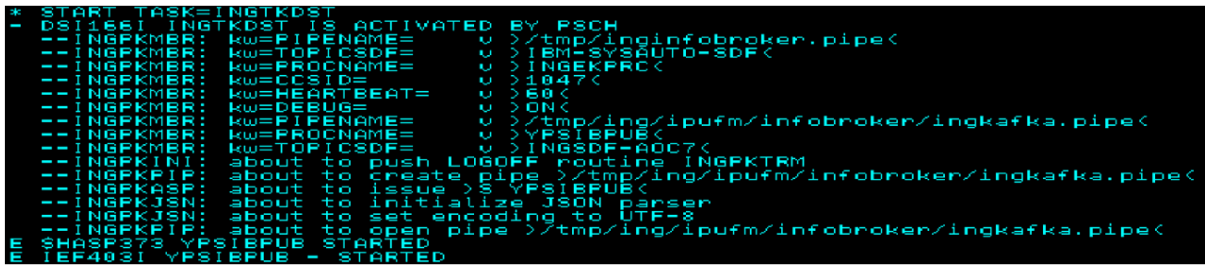
- Wrong procedure name, which is specified with the **PROCNAME** parameter in the INGKINIT initialization member.
- Unable to create the pipe, which is specified with the **PIPENAME** in the INGKINIT initialization member, because the pipe path is a read-only directory.
- Invalid encoding, which is specified with the **CCSID** parameter in the INGKINIT initialization member.

Solution

1. Enable debug in the initialization member INGKINIT.

   ```
   DEBUG=ON
   ```

2. Restart the task INGTKDST.
3. Check the Netlog for error messages.
4. Also, verify that the INGKINIT configuration that is printed in the Netlog is valid.



System Automation Info Broker startup problems

Possible error scenarios

- The configuration files directory (that is specified with the **IBRKR_CONFIG_HOME** parameter in file `ing.infobroker.environment`) was configured incorrectly.
- Files do not exist.
- Wrong permissions to read the files.
- The working directory of the System Automation Info Broker (that is configured with the **IBRKR_WORKING** property in file `ing.infobroker.environment`) is read-only.

```
ŸINFO"   - Ÿ File: ing.infobroker.security.properties "
ŸINFO"   - Attempt to convert the file
ŸINFO"   -    '/u/pisch/ing/infobroker/config/ing.infobroker.security.properties
ŸINFO"   - with charset IBM-1047 to file
ŸINFO"   -    '/tmp/ing/ipufm/infobroker/config/ing.infobroker.security.ascii.pr
ŸINFO"   - with charset ISO8859-1.
ŸERROR"  - File conversion has failed.
ŸERROR"  - Please ensure that the directory 'IBRKR_CONFIG_HOME' contains the fi
ŸERROR"  - IBRKR_CONFIG_HOME=/u/pisch/ing/Infobroker/config
ŸERROR"  - Please also ensure that the process/user has the necessary access pe

ŸERROR"  - IBRKR_WORKING=/tmp/ing/ipufm/infobroker/config
```

Solution

If you experience startup problems with the System Automation Info Broker, you can change the trace option in the JCL (default name is INGEKPRC) that is used to start the component.

1. Set the logging level of the JCL to trace '+T'.

   ```
   // LOGLVL='+T', <Debug LVL: +I(info) +T(trc)>
   ```

2. Restart the System Automation Info Broker.
3. Go to the end of SYSOUT and check for error messages. Error messages contain the error itself and an explanation of what caused this error.

JVM startup failure

Possible error scenarios

- Pipe was configured incorrectly.
- The directory for temp storage is read-only.
- Missing files or typos.

Solution

1. Locate the log file. The location of the log file is either configured in `ing.infobroker.logging.xml` or the default directory is used: `/var/log/ing/infobroker`.
2. Scroll to the end of the log file to check the error messages. Here is one example error.

   ```
   Caused by: java.io.FileNotFoundException Create breakpoint : /tmp/inginfobroker.pipe (Is a directory)
       at java.io.FileInputStream.open0(Native Method) ~[?:1.8.0_311]
       at java.io.FileInputStream.open(FileInputStream.java:195) ~[?:1.8.0_311]
   ```

JVM runtime exceptions

Possible error scenario: No messages are sent and received in the Apache Kafka topic.

Solution

1. Check if System Automation Info Broker Hub is running by executing the following command in NetView.

   ```
   LIST TASK=INGTKDST
   ```

   From the command output, you can tell if System Automation Info Broker is active or not.

   ```
   CNMKWIND OUTPUT FROM  LIST TASK=INGTKDST                    LINE 0 OF 8
   *------------------------------ Top of Data -------------------------------*
   TYPE: OPT TASKID: INGTKDST TASKNAME: INGTKDST STATUS: ACTIVE
   MEMBER: INGKINIT
   PRIMARY:NONE      STATUS:INACTIVE SECONDARY:NONE      STATUS:INACTIVE
   LOADMOD: DSIZDST
   Task Serial:   15407  REXX Environments:  0 (0%)
   Messages  Pending:   0   Held:   0
   WLM Service Class:   Not Available
   END OF STATUS DISPLAY
   *----------------------------- Bottom of Data -----------------------------*
   ```

2. If the System Automation Info Broker task status is not active, start it.
3. If the System Automation Info Broker task status is active, check for other reasons. Check the Java™ log for warning from the kafka-producer-network-thread. The default path of the log file is `/var/log/ing/infobroker/ing.infobroker.trace.log`.

If you notice the following messages, then either the Apache Kafka Broker is down or the property **`spring.kafka.bootstrap-servers`** (in file `ing.infobroker.properties`) was configured incorrectly.

```
[Producer clientId=sa-infobroker-producer-json-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Broker may not be available.
[Producer clientId=sa-infobroker-producer-json-1] Bootstrap broker localhost:9092 (id: -1 rack: null) disconnected
[Producer clientId=sa-infobroker-producer-json-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Broker may not be available.
[Producer clientId=sa-infobroker-producer-json-1] Bootstrap broker localhost:9092 (id: -1 rack: null) disconnected
```

Chapter 4. Starting and Stopping SA z/OS

This information contains the following sections:

Starting, Stopping, and Maintaining the Automation Manager

Normally, user interaction with the automation manager is limited to using ING*XXXX* commands to, for example, send the automation manager requests to start or stop resources. In special cases, however, there may be a need to interact with the automation manager during the initial start process and when it is necessary to stop an automation manager instance.

Starting the Automation Manager: START (S)

The automation manager runs in a z/OS address space of its own.

Normally, you would add this start command to the COMMND*xx* PARMLIB member so that the automation manager is automatically started at IPL time as described in *IBM Z System Automation Planning and Installation*. Alternatively, you can start it with an MVS start command that calls a module that initializes the automation manager.

In certain cases, you can also use the START (S) command.

Because the automation manager runs as a started task, the start command format may be modified by the installation. However a sample procedure is provided and can be used in most installations without modification. In addition to the various data sets that are identified by this procedure, there are several parameter values that may be specified. None of these parameters are required due to default values or additional sources of the information that are provided. You can find a sample startup procedure called INGEAMSA in the SINGSAMP sample library, see also *IBM Z System Automation Planning and Installation*.

In most cases, the parameters specified via the start command, if provided, override information that is obtained from the automation manager PARMLIB member or other default value. The only exception to this is that TYPE=HOT will be ignored when another automation manager instance has claimed primary mode.

Note: Do not start two instances of the automation manager with the same job name on the same system. They would come up as the primary and secondary automation manager. However, if you issue a stop command with the same job name, it would be accepted by both instances.

Also you cannot start more than nine instances of an automation manager per z/OS system within the sysplex or per single system.To allow your installation to specify a preference as to which automation manager instance should assume PAM responsibility at sysplex IPL, each automation manager enters a DELAY state. The duration of this state is set with the DELAY parameter of the automation manager initialization member of the PARMLIB (see also the description of the HSAPRM00 sample member in *IBM Z System Automation Planning and Installation*). This allows other instances with a lower delay value to assume PAM mode if started within the difference (in seconds) between the respective delay values. This makes sense when you would prefer that the primary automation manager runs on a special powerful processor.

Stopping the Automation Manager: STOP (P)

To allow an automation manager to be stopped or otherwise influenced by an operator, the automation manager "listens" for MVS MODIFY and STOP commands (see also "Making an Automation Manager the Primary" on page 102.) When an MVS STOP (P *jobname*) command is received, the automation manager

will be terminated. If the automation manager was the primary one, all work in process will be completed prior to actual termination.

If you try to stop the last primary automation manager, you will receive the reply HSAM1390E that you must confirm the automation manager stop request to.

Making an Automation Manager the Primary

You can use the GO subcommand of the MVS MODIFY (F) command to cancel the initial DELAY invoked when an automation manager instance is started thereby making it eligible to become the primary instance without having to wait for the delay time to complete.

This is useful when it is known that a preferred primary system will not become available during a sysplex wide IPL:

F *jobname*,GO

It is also possible to set the manager as the primary using the MVS MODIFY (F) command.

F *jobname*, SETMODE,PAM

Stopping the Automation Manager with Different Options

The STOP subcommand of the MVS MODIFY (F) command may be used to stop an automation manager instance in different ways:

F *jobname*,STOP,[NORMAL|IMMED|FORCE|DEFER],[NORESTART|RESTART]

The available options are:

NORMAL
The instance will be stopped after all in progress work has been completed and committed. No new work will be accepted from the automation agents, however any internally queued work will be processed. This is the default option.

IMMED
The instance will be stopped after the current work items have been completed and committed. Internally queued work may be lost.

FORCE
The instance will be stopped without waiting for in progress work to complete.

DEFER
The instance will be conditionally and normally stopped after the STOPDELAY interval has expired. Termination will occur only if another (secondary) instance is available, or if no automation agent is active.

On a secondary automation manager instance, the above variations are effectively treated as NORMAL as there is no work to be completed by SA z/OS. The second option specifies whether ARM restart processing should be attempted following the completion of the stop request.

NORESTART
The instance will be stopped as requested and no restart attempt will be made by ARM. This is the default option.

RESTART
The instance will be stopped in such a way as to cause ARM to attempt an automatic restart. Note that this will only occur if the instance is properly defined to ARM.

Shutdown of Primary Automation Manager (PAM) system

Provided a Secondary automation manager (SAM) is running in the sysplex, shutting down the PAM system is normally straightforward, with the SAM on an alternate system taking over the role of the PAM.

Refer to the section "Recovery Concepts for Automation Manager" in *IBM Z System Automation Planning and Installation*.

If however multiple system shutdowns or IPLs are initiated in the sysplex, it is occasionally possible to be left without an effective PAM. To avoid this, it is good practice to move the PAM to a stable system before initiating the shutdowns.

If the manager APLs are members of a MOVE or SERVER group, the INGGROUP command may be used with the EXCLUDE or AVOID parameters to facilitate the move (see *IBM Z System Automation Operator's Commands*, INGGROUP).

Obtaining Automation Manager Diagnostic Information

There are two methods of obtaining diagnostic information from the automation manager activities if problems occur:

- You can get a snapshot of the state image queue written into a specified data set using the command:

```
F jobname,DIAGINFO,SNAP,data_set_name
```

- You can get a log of the automation manager activities for problem reporting (maybe the ones to reproduce the problem). You issue a command to start the log that is written to a specified data set:

```
F jobname,DIAGINFO,RECORD,data_set_name
```

  With a second command, you terminate the log:

```
F jobname,DIAGINFO,ENDREC
```

Note: The DIAGINFO command accepts every data set name without checking whether the data set exists or is being accessed by some other user. If there are conflicts, the automation manager does not write any diagnostic information, but will not issue a message to the user.

Takeover File

The takeover file is dynamically allocated during the initialization of the primary automation manager (PAM). The name of the takeover file is defined in the HSAPRM*xx* parmlib member. The name of the takeover file cannot be changed when an automation manager (PAM or SAM) is running. All automation managers (in the same XCF group) must use the same takeover file name.

The takeover file is allocated using the DD name HSATKOVR. Note that an automation manager started as a SAM does not allocate the takeover file.

If an allocation error or a VSAM open/read error occurs during a hot start or takeover processing, the operator is prompted to decide whether to continue with a Warm start and lose all the information that is stored in the takeover file or to retry reading the takeover file. If the operator continues with a Warm start and this fails too, the automation manager terminates, which causes another takeover.

If an error occurs while writing to the takeover file, the automation manager continues to run using the in-storage version of the resources' objects but disables updating of the takeover file. The in-storage state information of the resources is not written to the takeover file when the automation manager terminates. You thus lose the ability to continue with a hot start when switching the primary automation manager.

The automation manager allows you to repair the takeover file while the PAM and the SAMs are running. To do this:

1. Disable the takeover file if this has not already been done by the PAM.
2. Repair the file, for example, by enlarging it.
3. Enable the takeover file again.

Starting and Stopping NetView

Starting NetView (System Operations)

NetView must be started on the host before you can use the system operations and processor operations of SA z/OS. NetView can be automatically initialized when you start z/OS.

For general information on initializing NetView, see *IBM Z NetView Installation: Configuring Additional Components*. You may have a dual NetView environment that consists of a Networking NetView (which runs NetView Graphic Monitor Facility and other networking facilities) and an Automation NetView (which runs SA z/OS). This topic assumes that both instances of NetView are initialized at the focal point system and that you are logged on to a NetView user ID at the SA z/OS workstation.

Your system programmer may have set up your system so that SA z/OS is started automatically when the Automation NetView initializes. (Directions for performing this setup are given in *IBM Z System Automation Planning and Installation*). As SA z/OS initializes, a multiline write-to-operator (MLWTO) message, AOF767I, is issued describing the available options. See Appendix B, “Automation Initialization Options,” on page 217 for the options that can be defined and for details about the MLWTO. To continue automation, enter:

```
R number
```

Where *number* is the message reply number of the accompanying AOF603D message. This is the default. If you do not issue a reply within two minutes, automation continues and SA z/OS loads automation policy from an automation control file. If Runmodes are defined in the policy with this reply, you could also set a runmode.

Note: Use the AOFINITREPLY advanced automation option (AAO) to suppress the issuing of the AOF603D message and have SA z/OS initialize with the default settings (see also the table "Global Variables to Enable Advanced Automation (CGLOBALS)" in *IBM Z System Automation Customizing and Programming*).

Stopping NetView

You can stop NetView using the **CLOSE** command with its various options. The option most commonly used is probably IMMED, which is degraded to STOP option when it is used on the very first **CLOSE** command.

SDF and GATEWAY definitions using non-XCF communication

If you have GATEWAY definitions in your active System Automation policy using non-XCF communication and you are using SDF, you should terminate NetView using the **CLOSE NORMAL** command first, followed by the **CLOSE** command that you normally prefer.

For details, refer to Gateway topic in *IBM Z System Automation Programmer's Reference*.

Starting and Stopping Processor Operations

1. Processor operations is started as an application running on the automation NetView at the focal point system. To start processor operations, issue the following command from a NetView operator console:

   ```
   ISQSTART ACF
   ```

   For more details about the ISQSTART command, see *IBM Z System Automation Operator's Commands*.
2. You stop processor operations by issuing **ISQSTOP** from the NetView operator console.

Chapter 5. Refreshing Automation Policy

This section describes how to identify the currently active configuration data set, how to refresh it using different ways, and how to handle configuration changes for processor resources.

Identifying Current Configuration Files

The automation configuration file is the file that controls system-level automation. It contains the SA z/OS automation policy, which specifies what, when, and how to automate. The NetView automation table (AT) specifies which automation procedure to run when various messages are received. The message revision table (MRT) enables user-defined modification of the attributes and text of original z/OS messages.

An automation configuration file is made available during SA z/OS initialization. The AT and MRT are loaded into NetView and the automation manager configuration file is loaded into SA z/OS.

Before refreshing the configuration file, you might need to identify the one that is currently active. The following sections describe how to identify the configuration file that is being used by an automation agent or the primary automation manager.

For an Automation Agent

You can use the DISPSYS command to identify the automation configuration file that is currently active. “Identifying Current Configuration Files” on page 105 shows an example of the information that is displayed.

```
Configuration
   Data set     : OPER1.USER.KEYPLEX.ACF(ACFZ995)
   Built by     : OPER1 04/11/04 10:20:36
   Activated    : 04/11/04 10:21:19
   CFG Token    : 200404051055068A0345169672

Message automation table(s)
    DSITBL01 INGMSGSA INGMSG01
```

Figure 25. DISPSYS Command Dialog

You can also use the details option of the INGAMS command. If you enter the command code **B** (Show Details) for an automation agent, it displays the DISPSYS output shown in “Identifying Current Configuration Files” on page 105.

For the Primary Automation Manager

To identify the current automation manager configuration file, you use the details option of the INGAMS command. On the INGAMS command dialog, you enter the command code **B** (Show Details) for a primary automation manager. A panel similar to Figure 26 on page 105 is displayed, which shows relevant information about the configuration files.

```
Config dataset name : OPER1.USER.KEYPLEX.ACF
Config member       : HSAZ999
                            Z999CLGC  STRUCTURE   20140405105506
                            Z999CRES  STRUCTURE   20140405105506
                            Z999CSCH  SCHEDULE    20140405105506
Config token        : 200404051055068A0345169672
Config version      : 01
```

Figure 26. INGAMS Command Dialog

You can also use the INGAMS command in line mode:

```
INGAMS DETAILS name OUTMODE=LINE
```

Where *name* is the name of the primary automation manager.

Refreshing Automation Policy using INGAMS command

The recommended method to refresh the SA z/OS automation policy is to issue the INGAMS command with the REFRESH option.

This command operates sysplexwide. All systems in the sysplex are refreshed automatically. It supports incremental updates on a system basis, that is, only the data that is affected by changes in the policy database is refreshed.

Other Ways to Refresh Automation Policy

Another way to refresh the automation policy is to use:

- The REFRESH subcommand of the MVS MODIFY (F) command, see "Refreshing an Automation Manager Configuration" on page 106

To reload the NetView automation tables that are specified in the System Info policy and the MRT, use the following command:

```
ACF ATLOAD
```

How the ATs and MRT are refreshed and reloaded depends on the setting of the AOFSMARTMAT advanced automation option (AAO). For more details, see its entry in the table "Global Variables to Enable Advanced Automation (CGLOBALS)" in *IBM Z System Automation Customizing and Programming*.

Refreshing an Automation Manager Configuration

The REFRESH subcommand of the **MVS MODIFY (F)** command may be used to initiate a configuration refresh operation, and is effective only when directed to the primary automation manager.

```
F jobname,REFRESH,new_data_set_name,[NOSUSPEND|FORCESUSPEND]
```

The *new_data_set_name* may be specified in several forms, as follows:

dsname
This form specifies the fully qualified data set name to be used.

***gdgname*(*generation*)**
This form specifies a generation data group name and the generation group member to be used. The generation number is relative to the most recently created member, and may be 0 (for the most recent) or a negative number (–1,–2, ..., –*n*) where *n* is the number of generations specified when defining the GDG, minus 1.

*****
This specifies that the refresh operation should use the same data set (and GDG member) as was used on the last successful refresh operation or during automation manager initialization.

***(*generation*)**
This specification form specifies that a member of the same GDG used for the previous refresh operation is to be used. The generation number is relative to the most recently created GDG member at the time of command entry.

The NOSUSPEND option ignores the processing of the whole suspend file.

The FORCESUSPEND option ignores all erroneous lines in the suspend file, but continues to process the suspend file.

If FORCESUSPEND is specified, or neither FORCESUSPEND nor NOSUSPEND is specified, the **F jobname,REFRESH** command will process the suspend file only if the configuration file has been changed as well. For example, if you have added, deleted, or modified a resources.

If no automation agent is running, you must use the REFRESH subcommand.

Example:

```
REFRESH,*(0) - use most recent generation of the specified GDG
```

Handling Configuration Changes for Processor Resources

This section addresses the advanced automation administrators and advanced operators who are already familiar with different SA z/OS product component details that are mentioned in this guideline.

This section summarizes how to activate PDB changes that are made for processor (PRO) entries at SA agent runtime. The processor configuration changes alone do not require to use the BUILD option ALL; option MODIFIED is sufficient and documented as follows. If you combine changes of PRO entries and other entry types, BUILD option ALL might still be needed.

Before activating the PDB changes on the SA agent side, resume the processor connections that are in a SUSPENDED runtime state but will be affected by the configuration change. It applies to the INTERNAL and SNMP connections only. It is meant to prevent deleted or renamed processor names from being kept in the DSISVRT VSAM Cglobal pool as SUSPENDED processors. Suspending processor connections in a production environment has important reasons. Therefore, plan and coordinate your processor change activations carefully in such situations.

Connection protocol specific information

- INTERNAL

  If the processor connections are exploited by GDPS, it's necessary to make sure that GDPS HMC Processor Connection Monitoring or automation scripts do not interfere with the planned processor configuration change action. SA does not perform its own connection monitoring in case a processor is running under GDPS control, according to the active GDPS definitions at runtime.

  When you add a new processor to your PDB, select entry type 37 AOP to make sure that enough HWOPERnn tasks are defined. Minimum task count is the number of mainframe processors configured for the INTERNAL interface plus 1.

  In addition to the AOP HWOPERnn task definitions, there must be corresponding AUTHW* task definitions in the DSIOPF operator include member AOFOPFA. A Netview/SA agent recycle is required to activate DSIOPF definition changes.

  If you need to rename the task names that are defined for the AOP HWOPERxx automated functions, special care must be taken to avoid any undesirable side effects. Active HWOPERxx tasks are running INGHWCOM permanently, therefore it is essential to stop any communication to any CPCs from any system in the current scope of the automation manager (the SAPlex) before the renamed tasks are activated as new HWOPERxx automated functions.

  Following steps should be taken in case you need to rename your HWOPERxx tasks:

  1. Define the new tasknames under AOP HWOPERxx, build a new ACF.
  2. Before activating the new ACF, stop communications to all CPCs from all systems in the SAPlex. Use INGHWSRV pro STATCOM to determine CPCs with active connections. You get reports displaying a STATUS of either CONNECTED or NOT_CONNECTED:

     AOFA0099 STATCOM pro STATUS(CONNECTED) TASK(AUTHWnnn)...

     AOFA0099 STATCOM pro STATUS(NOT_CONNECTED)...
  3. Ensure that the connections are not automatically reestablished, for example by instructing GDPS to not touch the connections now. See GDPS documentation.
  4. For each CPC having a status of CONNECTED, use INGHWSRV pro FORCE to terminate the connection. You get a report about the success of the operation:

     AOFA0004 TERMCOM pro STATUS(SUCCESS)...

5. If there are no active connections left, activate the new ACF using INGAMS REFRESH. Check whether the new autotasks are active for the HWOPERxx automated functions by using the DISPAOPS command.
6. Stop any of the old autotasks that are still running using the NCCF LOGOFF command:

   /AUTHWnnn: LOGOFF

   As they are no longer part of the active configuration no automatic restart should occur.
7. Check whether CPC connections is reestablished. If not, use INGHWSRV pro CONNECT to manually reestablish. You get a report about the success of the operation:

   ING824I your_userID requests to connect to CPC pronet.pro...

   AOFA0002 INITCOM pro STATUS(SUCCESS)...
8. Enable previously disabled CPC operations, for example let GDPS handle and monitor the established connections again.

- SNMP, TCP/IP

  Since any change in its data model cannot be activated while ProcOps itself is up and running, ProcOps needs to be recycled to pick up the SOCNTL/ACF changes into its data model. Recycling ProcOps is disruptive for all its active processor connections, regardless if there was a configuration change for them or not.

  When you add a new processor, make sure that your DSIOPF include member AOFOPFPO contains an adequate number of ISQCMxxx and ISQBTxxx task definition statements. The minimum task definition number of each task type is the sum of all SNMP and TCP/IP processor connections. A Netview/SA agent recycle is required to activate DSIOPF definition changes.

Steps to activate processor policy changes

- Delete a PRO entry
  - INTERNAL connection

    If you have configured a rather short connection monitoring interval (less than 10 minutes) for a processor you want to delete, it is recommended to change the monitoring interval to NONE in an extra PRO modify step before the actual PRO delete operation. It prevents the monitoring from inadvertently restarting the connection after it has been already stopped as part of the delete procedure.

Step	Action
1	PDB: Perform the PRO entry DELETE actions.
2	PDB: Do a BUILD MODIFIED.
3	SA agent: Issue the **INGHWSRV pro FORCE** command, where *pro* is the name of the deleted processor. Repeat this command on each system in your SAplex, which has a connection to *pro* and share the changed SOCNTL/ACF.
4	SA agent: Issue the **INGAMS REFRESH CFG=*** command. SA will trigger the REFRESH on the other systems in the SAplex, sharing the same SOCNTL/ACF. Active connection monitor timers of deleted processors will be purged automatically.
5	SA agent: Issue the **INGCLEAN** command. Repeat this command on each system in your SAplex, sharing the changed SOCNTL/ACF.

  - ProcOps SNMP/TCPIP connections

Step	Action
1	PDB: Perform the PRO entry DELETE actions.
2	PDB: Do a BUILD MODIFIED.
3	SA agent: Issue the **INGAMS REFRESH CFG=*** command. SA will trigger the REFRESH on the other systems in the SAplex, sharing the same SOCNTL. Active connection monitor timers of deleted processors will be purged automatically. Repeat this command on each system in your SAplex, sharing the changed SOCNTL/ACF.
4	SA agent: Issue the **ISQSTOP** command to end Procops. Issue the **ISQSTART ACFCLEAN** command to erase the old configuration and restart Procops with the changed configuration. Repeat this step on all Procops instances that are active in your SAplex, sharing the changed SOCNTL/ACF.

- Modify or replace an existing PRO entry
  - INTERNAL connection

Step	Action
1	PDB: Perform PRO entry rename and/or policy change actions.
2	PDB: Do a BUILD MODIFIED.
3	SA agent: Issue the **INGHWSRV pro FORCE** command, where *pro* is the name of the modified or replaced processor. Repeat this command on each system in your SAplex, which has a connection definition to *pro* and share the changed SOCNTL/ACF.
4	SA agent: Issue the **INGAMS REFRESH CFG=*** command. SA will trigger the REFRESH on the other systems in the SAplex, sharing the same SOCNTL/ACF. Active connection monitor timers of any replaced processor will be purged automatically.

  - ProcOps SNMP/TCPIP connections

Step	Action
1	PDB: Perform PRO entry rename and/or policy change actions.
2	PDB: Do a BUILD MODIFIED.
3	SA agent: Issue the **INGAMS REFRESH CFG=*** command. SA will trigger the REFRESH on the other systems in the SAplex, sharing the same SOCNTL/ACF. Active connection monitor timers of replaced processors will be purged automatically.
4	SA agent: Issue the **ISQSTOP** command to end the ProcOps function. Issue the **ISQSTART ACFCLEAN** command to call INGCLEAN and restart ProcOps in one go with the changed configuration. Repeat this step on all ProcOps instances that are active in your SAplex, sharing the changed SOCNTL/ACF.

- Add a new PRO entry
  - INTERNAL connection

Step	Action
1	PDB: Perform NEW PRO entry actions.

2	PDB: Do a BUILD MODIFIED.
3	SA agent: Issue the **INGAMS REFRESH CFG=*** command. SA will trigger the REFRESH on the other systems in the SAplex, sharing the same SOCNTL/ACF.

– ProcOps SNMP/TCPIP connections

Step	Action
1	PDB: Perform NEW PRO entry actions.
2	PDB: Do a BUILD MODIFIED.
3	SA agent: Issue the **INGAMS REFRESH CFG=*** command. SA will trigger the REFRESH on the other systems in the SAplex, sharing the same SOCNTL/ACF.
4	SA agent: Issue the **ISQSTOP** command to end the ProcOps function. Issue the **ISQSTART ACFCLEAN** command to clear the old configuration and restart ProcOps with the changed configuration. Repeat this step on all ProcOps instances that are active in your SAplex, sharing the changed SOCNTL/ACF.

Part 2. Using SA z/OS on the Host

This part describes how to use SA z/OS in a host session. It contains the following chapters:

Chapter 6. Getting Started

Different sets of commands are available from host sessions:

- Automation manager commands
- System operations commands (automation agent commands)
- Processor operations commands

With system operations commands, you can control and maintain all of the resources sysplexwide from a single point of control. They operate in two modes:

- Fullscreen mode: If it is a command to display information and you just enter the command name, a panel is displayed showing all available resources. On this panel, you can specify further actions/ commands for a special resource.

  If you enter a command to maintain a resource without further parameters, a fullscreen panel prompts you for more information.
- Line mode: From a command line, you can enter the complete syntax of a command to receive either the desired output directly or to manipulate the resource you wanted in the way you wanted.

All system operations commands (automation agent commands) also operate in fullscreen mode and line mode unless stated in the reference documentation that the OUTMODE parameter is not supported.

The tasks that you can manage with the available commands are described in more detail in the subsequent chapters and sections. If you need the complete syntax of these commands, refer to *IBM Z System Automation Operator's Commands*.

Issuing Commands

With SA z/OS, there are several ways that you can issue a command:

- You can use command dialogs to enter command parameters and issue commands using 3270 panels. See “Using Command Dialogs” on page 113.
- You can issue processor operations, system operations, and automation manager commands from the same interface via NetView terminals as described in “Using Command Dialogs” on page 113.

For each method, instructions on the window or panel tell you what to do. The commands are described in greater detail in *IBM Z System Automation Operator's Commands*.

Using Command Dialogs

Enter A0C from the NetView command line in your NetView session to reach the System Operations MAIN MENU.

Enter 2 or CD at the command line to invoke the Command Dialogs Main Menu. Starting with this menu, you can interact with 3270-type panels to issue SA z/OS system operations commands.

You can enter a command with two methods:

1. Type the number or letter corresponding to the desired command at the command line,
2. Type the name of the command. If you choose to type a command, you can add options to it.

For example, if you enter number 2 at the command line of the Command Dialogs Main Menu, you reach the AOCTRACE command dialog that prompts you for further options.

Some of the commands operate sysplexwide; especially those sent to the automation manager. For other commands you can specify a target system that they should operate on, for example, if you want to display the automation flags on a certain target system.

The commands present a snapshot of the data, taken at the time the command is executed. Some data may be refreshed when other commands are issued from the same panel. To force a refresh of all data you must press PF9.

Command Logging

SA z/OS allows you to log the commands and their parameters that are issued by an operator. You can then use this log for audit purposes.

Several operator commands write an AOF705I message for each command execution. The AOF705I message lists all the parameters that have been specified together with the user ID of the operator or autotask that issued the command, for example:

```
AOF705I USER1 issued command INGREQ with MYAPL/APL/SAT1 REQ=START SCOPE=ONLY
        VERIFY=YES SOURCE=OPERATOR PRI=LOW PRECHECK=YES

AOF705I USER1 issued command INGIMS with IMS3CTL/APL/KEYC REQ=CMD CMD=/DIS
```

You can enable and disable command logging using the INGCNTL command, which sets the AOF_AAO_LOG_COMMAND advanced automation option (AAO), for example:

```
INGCNTL SET COMMAND_LOGGING=YES
```

Chapter 7. How to Get Resource Information Sysplexwide

This information describes commands that you can use to retrieve information from all resources that are defined within a sysplex from a single point of control.

The commands are presented here in an overview. Each command is documented with sample screens and the complete syntax in *IBM Z System Automation Operator's Commands*.

SA z/OS provides commands that let you examine resources from the point of view of the automation manager or automation agent.

The automation manager decides how automation is performed for known resources in the sysplex. The automation manager decides how to handle the requests that it receives through the various channels (for example, the defined automation policy or interactive operator requests) using an internal hierarchy of priorities that is given in Table 3 on page 12.

SA z/OS offers a series of commands that you can use to find out what information the automation manager currently holds about resources, such as relationship information or information about the status of resources and whether schedules exist for them.

An automation agent is installed on each system with resources that need to be automated by SA z/OS. For example, automation agents are responsible for:

- Having the appropriate commands processed for subsystem startup, shutdown and suspension
- Reporting status updates to the automation manager
- Message automation

Automation agents access information for their tasks from the automation configuration file.

SA z/OS offers commands that you can use to acquire information about resources running on the system where the automation agent resides.

Using INGLIST to View Resources

The INGLIST command provides details about resources in the sysplex from the point of view of the automation manager. It displays information about a resource, such as statuses, flags, and schedules.

On the INGLIST panel, you can:

- See more information about the resources by scrolling horizontally using PF11.
- Use PF9 (Refresh) to obtain a new set of data for the displayed subsystems.
- Use PF5 to invoke a filter dialog (as described in “Specifying Filters with INGFILT” on page 116), which is equivalent to using the INGFILT command that is available from any command line. With this filter, you can reduce the amount of information shown in the display. It enables you to setup a specific view for an operator. For example, you can set up a filter that shows only certain resources in specified observed and desired states.
- Launch other actions to maintain and control resources or groups.
- Use the **CMD** column on the left of the panel to issue various commands against any of the resources that are displayed.

  The available commands are indicated at the top of the INGLIST panel:

```
  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
```

Note: If you have defined exit routine AOFEXC04, there is an additional action code U. See *IBM Z System Automation Customizing and Programming* for details.

Enter the letter corresponding to the command in the entry field next to the resource you want it issued against and press Enter. When you return from the command there may be a slight pause as the display refreshes itself.

- Use PF4 to toggle to the DISPSTAT command dialog, which gives you the automation agent view of the resources in the sysplex (see "Displaying the Automation Agent View with DISPSTAT" on page 118). To return to the INGLIST panel press the PF3 key on the DISPSTAT panel.

  Pressing PF4 on the DISPSTAT panel displays the INGLIST panel for all resources that are shown on the DISPSTAT panel. However, this might not be the same as the original INGLIST display because the DISPSTAT display might have been modified due to filtering or suppressing of resources that the automation agent does not handle.

For more detailed information, see *IBM Z System Automation Operator's Commands*.

Specifying Filters with INGFILT

You can use the INGFILT command dialog to specify or revise filter criteria. The filter settings are saved in task global variables, so that when you next run INGLIST they will be used again. Press PF9 to save the currently displayed filters in these task global variables.

You can specify one or more of the following filters:

- Resources
- Observed status
- Desired status
- Automation status
- Compound status
- Health status
- Automation flag
- Category
- Subcategory
- Group type
- Jobname(s)
- Description
- RunToken(s)
- Pacing Gate(s)

You can abbreviate the status in the status lists, for example, av for Available. In order to be eligible for the display, a resource must match all filter criteria. An asterisk means that the filter is not set. A '^' or '\' in front of the status displays all resources whose status is not what you specified.

Use the PF4 key to clear the currently established filter settings. The filter will then contain an asterisk, meaning that the filter is not set, or a blank. Use the PF5 key to revert to the currently established filter settings.

Note: If INGFILT was called from INGLIST, the filters that are displayed will take effect when you press Enter. If you press PF9 first, the filters will be saved.

Using INGINFO to View a Selected Resource

The INGINFO command displays details about a specified application or application group from the viewpoint of the automation manager. If you enter this command without parameters, a dialog prompts you for more information.

For a detailed description of the command syntax refer to *IBM Z System Automation Customizing and Programming*.

You can also issue this command against any resources that are displayed in the result panel of the INGLIST command. In the resulting panel you see information about the resource's different statuses, automation flag settings, whether dependencies are satisfied and what votes are currently active against the resource.

On the INGINFO panel, you can use PF4 to toggle to the DISPINFO command dialog, which gives you the automation agent point of view of the current resource (see "Using DISPINFO to Display Detailed Information" on page 119).

If the resource has pacing gates defined, you can use PF5 to toggle to the INGPAC command dialog, which gives you more details about the pacing gates.

Displaying Application Group Information

The DISPAPG command displays detailed information about a specified application group. If you do not specify a application group, a menu is displayed where you can specify the application group name.

Use PF9 to refresh the information about the current application group and PF4 to invoke the INGINFO command to get details about the application group from the automation manager's point of view.

Displaying Monitor Information

The DISPMTR command displays information about monitors that you have defined using the customization dialog for your system and allows you to manage them (that is, resetting the health status of the resource that is being monitored to NORMAL, and starting and stopping monitors).

Displaying Relationship Information with INGRELS

The INGRELS command displays a panel that shows the relationships that are defined for the specified resource and the status of the current conditions (whether they are satisfied or unsatisfied).

The INGRELS command dialog shows the relationships that have been defined for the current resource in both directions, so in the **Dir** column from the INGRELS command dialog there is:

- A **B** for backward relationships, that is, the relationship exists from the partner resource to the specified resource
- A **F** for forward relationships, that is, the relationship exists from the specified resource to the partner resource

Relationship conditions are also shown (with the signature `Cond:`) and whether they are unsatisfied.

You can change the level of detail that is displayed using the following keys:

+
: Shows the condition details that exist for the selected relationship. This is only applicable if the relationship conditions were previously collapsed.

–
: Collapse the condition details. Only the relationship will be shown.

Displaying the Automation Agent View with DISPSTAT

The DISPSTAT command displays fullscreen panels that show information for resources defined to the current automation agent, such as automation status, automation flag setting overrides, and automation status changes.

Status conditions are color-coded to make it easier to identify problems.

On the DISPSTAT panel, you can:

- Cycle through further information using the PF10 and PF11 keys. If you use a wide (132 column) display, all of the data can be displayed on two screens. On a smaller 80 column display there is a third screen that displays Description data.
- Use PF9 (Refresh) to obtain a new set of data for the displayed subsystems. Some of this information, especially the automation status, changes in real time.
- Use PF5 to invoke a filter dialog (as described in "Specifying Filters with DISPSFLT" on page 118), which is equivalent to using the DISPSFLT command that is available from any command line. With this filter, you can reduce the amount of information shown in the display. It enables you to setup a specific view for an operator. For example, you can set up a filter that shows only resources of a certain name or that are in a specific state.
- Use the **CMD** column on the left of the panel to issue various fullscreen commands against any of the resources that are displayed.

  The available commands are indicated at the top of the DISPSTAT panel:

```
  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
```

  Note: If you have defined exit routine AOFEXC04, there is an additional action code U. See *IBM Z System Automation Customizing and Programming* for details.

  Enter the letter corresponding to the command in the entry field next to the resource you want it issued against and press Enter. When you return from the command there may be a slight pause as the display refreshes itself.
- Use PF4 to toggle to the INGLIST panel to see the same resources as shown in the DISPSTAT panel but from the automation manager point of view. To return to the DISPSTAT panel press the PF3 key on the INGLIST panel.

  Pressing PF4 on the INGLIST panel displays the DISPSTAT panel for all resources shown in the INGLIST panel. However, this might not be the same as the original DISPSTAT display because the INGLIST display might have been modified due to filtering.

For more detailed information, see *IBM Z System Automation Operator's Commands*.

Specifying Filters with DISPSFLT

The DISPSFLT panel displays the filters that are used for the DISPSTAT display. The filters are stored in task global variables, and you can save and update them by pressing PF9.

When you invoke the panel from DISPSTAT, pressing Enter takes you to the DISPSTAT display using the filters that are currently displayed. The filters are not saved unless you have previously pressed PF9 in the DISPSFLT command dialog.

The filters that you can set are:

- **Resources:**

  You can specify a list of one or more subsystem names. They can have leading or trailing wildcards (for example, *VIEW, NET* and *TVIE* are okay, but NE*EW will not work).

You can also specify a list of job names instead of subsystem names.

- **Statuses:**

  You can specify a list of automation statuses to be included or excluded from the display. This filter is used whenever the display is refreshed.

  SA z/OS provides the following sets of pre-grouped statuses, which can be abbreviated by their numbers:

  1. All resources that are currently UP or ENDED
  2. All resources that are in a normal status: DOWN, STARTED, UP, AUTOTERM, AUTODOWN, ACTIVE, ENDED, ENDING, RESTART, EXTSTART, RUNNING
  3. All resources that are in a down status: DOWN, INACTIVE, RESTART, AUTODOWN, CTLDOWN, STOPPED, BROKEN, MOVED, FALLBACK
  4. All resources that are in a start transition state: RESTART, STARTED, STARTED2, ACTIVE, RUNNING, EXTSTART
  5. All resources that are in a stop transition state: AUTOTERM, ENDING, STOPPING, ABENDING, BREAKING

- **Target Systems:**

  This lets you specify the systems within the sysplex that you want data gathered from. By default it shows just your local system.

  You can specify a list of one or more system, domain, or sysplex names (in any combination), and the query is sent to those systems. Alternatively you can specify '*ALL', and the query is sent to all active systems within the sysplex.

  In general the more specific your query and the fewer systems you send it to, the faster it will execute.

  When you have updated the filters (the defaults will do for just exploring), press Enter to go into the DISPSTAT display.

Using DISPINFO to Display Detailed Information

To see detailed information on a subsystem from the automation agent point of view, use the DISPINFO command. This will only work on one subsystem at a time. If the subsystem you specify happens to be on more than one system within the sysplex, you will be prompted to select the one you would like to see information about.

The DISPINFO command displays information about the selected resource including many of the definitions that have been made to SA z/OS about the subsystem. You can use PF4 to toggle to the INGINFO command display for the selected resource, which shows the automation manager view of it.

You can use PF8 to scroll forward through the pages of information, and PF7 to scroll back.

Use PF8 until you reach a display line like the following that indicates the end of the display:

```
*** Data produced at 17:58:17 on 07/26/05.
```

If you attempt to scroll further down than the last page, an error message is displayed:

```
AOF169I THE LAST LINE IS ALREADY DISPLAYED ON THE SCREEN
```

Note: If you have defined the exit routine AOFEXC03, the DISPINFO panel also shows user supplied information about the subsystem. Refer to *IBM Z System Automation Customizing and Programming* for more information about AOFEXC03.

If you use DISPINFO for a CICS-controlled or IMS-controlled resource, PF10 is active and with this you can view additional CICS or IMS related information as if you issued the INGCICS REQ=INFO or INGIMS REQ=INFO commands.

DISPINFO shows further information for resources that have been defined with INGVTAM startup and shutdown commands. It displays the specific APPLID and optionally the ACB (if it is different from the APPLID) that the resource is using. It also shows the generic APPLID and USERVAR if any have been defined. Finally a list of major nodes that have been registered as being used by the resource is displayed.

Using DISPTREE to Display Dependency Information

Generally if a resource has not been started, it will be because of a problem with one of its supporting resources. You may also wish to know which other subsystems will be impacted by problems with a particular subsystem. To find out what the problem is, use the DISPTREE command.

The DISPTREE command presents a view of the start dependencies, stop dependencies or the group membership of the specified subsystem (resource). Supporting resources are shown above and to the left of the specified subsystem, its dependent resources are shown below and to the right.

The panel uses reverse video to improve the highlighting. Non-subsystem resources are indicated with |.

PF4 will refocus the display onto the subsystem under the cursor. PF5 will invoke DISPINFO against the subsystem under the cursor. However, you cannot use DISPINFO on non-subsystem resources.

Displaying Statistical Information

The INGRPT command displays a panel providing statistical information about the automation agent and some basic information about the automation manager of a system in a sysplex.

The summary report lists information about, for example:

- The number of resources defined to NetView and managed by the automation agent
- The total number of messages automated
- The total number of resulting commands
- The total number of START and STOP commands issued
- The number of systems in the sysplex
- The total number of resources managed by the automation manager

If you want to see detailed statistical information for a particular resource, issue the command INGRPT STATS=DETAIL. A panel is displayed that shows statistical information about all resources that are automated by the automation agent.

For each resource the following information is shown:

- The number of messages automated
- The number of commands resulting from the message traps
- The number of START and STOP commands that were issued
- The number of critical threshold conditions that have occurred

You can select a resource to display details by entering S in the Cmd column.

Chapter 8. How to Monitor and Control Resources

This information introduces commands that you can use to control resources. Controlling resources in this context means making them available or unavailable.

In addition to the principal methods for controlling resources as introduced in Chapter 3, “Concepts,” on page 9, SA z/OS offers more sophisticated techniques to the operator.

With the commands introduced in this chapter, you can, for example:

- Start or stop a resource or application group
- Suspend and resume automation for a resource or application group
- View resource or group details
- Turn on or off the automation flag for a resource or application group
- View and modify service periods, referred to as schedules
- Manage sysplex resources
- Discover automated resources and their relationships within a sysplex
- Display resource statistics
- View trigger definitions

Table 15 on page 121 shows the various tasks that are discussed in this chapter along with the main commands that you can use to perform them. For more detailed information about the commands refer to *IBM Z System Automation Operator's Commands*.

Table 15. Operator Tasks and Related Commands

Task	**Commands**
“Starting, Stopping and Suspending Resources” on page 122	INGREQ, INGVOTE, INGSUSPD
“Application Pacing” on page 125	INGPAC
“Using Runmodes” on page 127	INGRUN
Displaying history information, “Using INGHIST” on page 124	INGHIST
“Using Schedules” on page 133	INGSCHED
“Displaying and Setting Trigger Information” on page 134	DISPTRG, INGTRIG
“Displaying and Setting Events” on page 134	DISPEVT, DISPEVTS, INGEVENT
“Determining Automation Agent Flag Values” on page 135	DISPFLGS, DISPSCHD, DISPSTAT, INGAUTO
“Working with Application Groups” on page 137	INGGROUP, DISPAPG
“Moving Sysplex Application Groups” on page 140	INGMOVE
“Setting Timers” on page 141	SETTIMER
“Interacting with the Automation Manager” on page 141	INGAMS
“Changing the Automation Agent Status” on page 142	SETSTATE
“Monitoring Messages” on page 142	INGMSGS, INGNTFY, ISQCMON, ISQXMON
“Monitoring Hardware” on page 145	ISQXDST
“Monitoring Health Performance” on page 146	DISPMTR, INGLIST

Table 15. Operator Tasks and Related Commands (continued)	
Task	**Commands**
“Monitoring OMEGAMON Sessions” on page 147	INGSESS
“Monitoring Processor Status” on page 150	ISQXDST
“Monitoring with the Status Display Facility” on page 153	SDF
“Monitoring and Controlling a Sysplex” on page 163	INGPLEX, INGSTR
“Monitoring and Controlling Logical Partitions and Guest Systems” on page 179	ISQESUM, ISQCCMD
“Displaying Threshold Settings and Occurrences” on page 190	DISPASF, DISPERRS, INGTHRES
“Handling Jobs That Are Not Controlled by SA z/OS” on page 193	INGLKUP
“Communicating with Other Systems” on page 193	INGSEND, ISQSEND

Starting, Stopping and Suspending Resources

In addition to goal driven automation that you exploit by defining automation policy, SA z/OS offers a series of commands that let an operator react to special circumstances beyond regular automation, including:

- INGREQ, see “Using INGREQ” on page 122
- INGSUSPD, see “Using INGSUSPD” on page 123
- INGVOTE, see “Using INGVOTE” on page 124
- INGHIST, see “Using INGHIST” on page 124

Each of these commands is documented with complete syntax and further sample output screens in *IBM Z System Automation Customizing and Programming*.

Using INGREQ

As described in “Automation Concepts” on page 9, commands issued with INGREQ and a high priority are the most powerful automation commands to make applications available or unavailable. They supersede all other desired status requests that exist against a resource at that time. Suspend requests against a resource can only be overruled with an INGREQ command and OVERRIDE=SUS.

You can use this command to:

- Initiate the start process of one or more specified application resources or application group resources
- Initiate the shutdown process of one or more specified application resources or application group resources

A desired status request is persistent until:

- You cancel it using the INGVOTE, INGREQ, or INGSET command
- It expires according to the specified EXPIRE parameter
- It is automatically removed when a specified condition comes true
- You want it to be cancelled after a specified timeout period, if the request has not been satisfied after that time
- It is overwritten by a desired status request from the same source

Even though desired status requests are persistent, it is not guaranteed that they will be satisfied by the automation manager.

With a desired status request you specify two categories of parameters:

- How should the request be executed (for example, with low or high priority)?
- What should happen with the request during its lifetime, for example, should it expire after a certain time anyway, or after a timeout if it has not been successful?

If you specify the command name without any parameters, SA z/OS displays a panel where you can specify, for example, startup or shutdown parameters, a priority, and a timeout for the desired status request.

Use PF11 to see additional parameters.

After pressing Enter to submit the desired status request, you will see a verification panel that lists all resources that are affected either when posting an INGREQ command to the automation manager or when cancelling a previously made desired status request. When making a desired status request, a resource is affected if it is within the dependency graph that the request is propagated along. This will also display resources that are already in the desired state. In this case nothing will happen to the resource.

When cancelling a desired status request, a resource is affected when it holds a vote for the request to be removed. Only resources that are highlighted will be processed by the command. All other resources are already in the state satisfying the appropriate dependencies.

Starting Subsystems

To start a subsystem it must be put into either the DOWN or RESTART automation status that is seen by the automation agent. The DOWN status occurs just after SA z/OS initialization and indicates that the subsystem has not been active since SA z/OS was last started.

To put a subsystem into the RESTART status, you can use the SETSTATE command.

Using INGSUSPD

As described in “Automation Concepts” on page 9, commands issued with INGSUSPD can not be overruled by a command issued with INGREQ. They can only be overwritten with the OVERRIDE parameter of INGREQ.

You can use this command to:

- Suspend automation of one or more specified application resources or application group resources
- Resume automation of one or more specified application resources or application group resources

A suspend request is persistent until:

- You cancel it using the INGVOTE, INGSUSPD, or INGSET command.
- It expires according to the specified EXPIRE parameter.
- You want it to be canceled after a specified timeout period, if the request has not been satisfied after that time.
- It is overwritten by a suspend request from the same source.

Even though suspend requests are persistent, it is not guaranteed that they will be satisfied by the automation manager.

With a suspend request you specify two categories of parameters:

- How should the request be executed (for example, with a scope only or a scope all)?
- What should happen with the request during its lifetime. For example, should it expire after a certain time anyway, or after a timeout if it has not been successful?

If you specify the command name without any parameters, SA z/OS displays a panel where you can specify, for example, suspend parameters, a scope, and a timeout for request.

After pressing Enter to submit a suspend request, you will see a verification panel that lists all resources that are affected either when posting an INGSUSPD command to the automation manager or when canceling a previously made suspend request. When making a suspend request, a resource is affected if it is within the dependency graph that the request is propagated along. This will also display resources that are already suspended. In this case nothing will happen to the resource.

When canceling a suspend request, a resource is affected when it holds a vote for the request to be removed. Only resources that are highlighted will be processed by the command. All other resources are already in the state satisfying the appropriate dependencies.

Using INGVOTE

Whenever you (or a program) want the automation manager to do something, you must send it a request. Requests are remembered within the automation manager. They are made against individual resources and then propagated across the relationship graph to other resources within the automation manager. If, for example, your request is to start resource A, and A has a relationship to B such that B needs to be started before A can run, the automation manager generates a vote to resource B to start it. Thus a request can be viewed as a black box of related votes that is generated by the automation manager (desired status requests and suspend requests).

If you enter the INGVOTE command without any parameters, it displays all pending requests that are persistent at the automation manager.

If you enter the INGVOTE command specifying a resource, or launch it from the INGLIST command dialog, SA z/OS displays all pending votes for that resource with the following information:

- The name of resource the request was made against
- The request action
- The source of request
- The request priority
- The request vote
- Default desired status
- Runmode qualification

From the INGVOTE command dialog, you can enter various commands against a selected vote.

Using INGHIST

The INGHIST command is a sysplexwide command issued against the sysplex's automation manager. It displays history information about the work items processed by the automation manager. You can specify a certain resource that you want SA z/OS to display the work history for, or you can specify a time interval that you want to see all activities for, or you can combine both criteria. Additionally you can specify the maximum number of work items to be shown. Use the WIMAX parameter to:

- limit the output if many work items exist within the specified time period
- specify the maximum number of work item records to be shown after work item expansion in fullscreen mode.

Time-interval search criteria are entered in local time. However, because the history data is stored using GMT time stamps, it is necessary for INGHIST to convert the local-time search criteria to GMT before retrieving the data. This may make it necessary for the operator to make a manual adjustment to the time-interval search criteria after a daylight-saving-time change has been set. The local-time-to-GMT conversion calculation will be one hour different after a daylight-saving-time change. Data written after a daylight-saving-time change will be retrieved as expected but data written before will be offset by one hour.

If you enter the INGHIST command without parameters, you receive the output for the complete sysplex for the period starting from the last hour before you entered this command.

In the **Cmd** column on the left of the INGHIST panel, you can expand each work item with the **+** or **e** command to see information about the detail actions that have been performed by the automation manager and what resources were involved.

You can collapse the expanded work items again with - or c.

Application Pacing

The resource consumption of applications (that is, of type APL) during the start and stop phases can vary a lot. While many applications are started or stopped very quickly, other applications may consume a lot of CPU resources and may even tend to dominate how the system assigns CPU resources among the started tasks in need.

To facilitate the management of those applications aiming for efficient resource utilization while avoiding customization efforts, the application pacing capability can be used to control how many applications of a kind can be started or stopped at the same time. For this, any application that is eligible to receive a start or stop order from the automation manager has to 'transit' through a defined Pacing Gate. If the number of applications that are currently in transition reaches the maximum concurrency level, additional applications are held back (waiting) until another application finished this transition and reached the final desired status or terminated during that transition.

In order to use application pacing, you need to define one or more Pacing Gate entries using the Customization Dialog, option 13 (PAC) and specify the concurrency level of starting and stopping applications. See Figure 27 on page 125.

```
                            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
```

Figure 27. Entry Type Selection Panel

Refer to *IBM Z System Automation Operator's Commands* for further details about creating Pacing Gates and the options provided.

Next, you assign those applications that you want to more tightly control to a Pacing Gate. This is done by selecting a pacing gate using the PACING GATE policy. Note, each application can only be associated with one single Pacing Gate at a time.

Typically, application start-up and shutdown times can be minimized if you allow for maximum parallelism that the system can handle. Therefore, it is not recommended to use Pacing Gates as a general vehicle to control all applications. Using them for certain CPU intensive workloads, however, can help to decrease system IPL, shutdown times or both, as Pacing Gates can prevent too many workloads dominating the CPU resources during these times.

To see how Pacing Gates affect the applications associated with them, invoke the INGPAC command from the NetView console. INGPAC shows a list of Pacing Gates based on the selection criteria specified by the

operator. It also allows you to display all applications that are currently either in transition or waiting for transition through the Pacing Gate. An example display is shown in Figure 28 on page 126:

```
INGKYPA0             Automation Control - Command Dialogs  Line 1     of 10
Domain Id . : IPUFL   ------------ INGPAC   ------------   Date . . : 07/11/14
Operator Id : JMH             Sysplex = AOC6PLEX           Time . . : 15:38:14

CMD: D Details L List Resources

CMD Pacing Gate          Type  System   Limit    Num Res  Num Wait Avg Wait
--- -------------------- ----- -------- -------- -------- -------- --------
__  AWASGATE            Start AOC6            2        0        0        4
__  AWASGATE            Stop  AOC6            3        0        0       12
__  WASGATE             Start AOC6            2       18       16      491
__  WASGATE             Stop  AOC6      NOLIMIT        0        0        -
__  WASGATEA            Start AOC6            1        0        0        -
__  WASGATEA            Stop  AOC6      NOLIMIT        0        0        -
__  WASGATEB            Start AOC6            1        0        0        -
___ WASGATEB            Stop  AOC6      NOLIMIT        0        0        -
___ WASGATEC            Start AOC6            1        0        0        -
___ WASGATEC            Stop  AOC6      NOLIMIT        0        0        -


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

Figure 28. INGPAC Command Dialog Panel

Refer to *IBM Z System Automation Customizing and Programming* for further details about using INGPAC to manage Pacing Gates.

Usage notes:

1. Performing a configuration refresh while resources are being started or stopped under the control of a Pacing Gate may cause the resources to be released immediately:

   - if you remove a resource from a Pacing Gate
   - if you remove a resource from a Pacing Gate and link it to another Pacing Gate

   For these reasons it is not recommended to perform a configuration refresh while resources are being started or stopped under the control of pacing gates.
2. While you may use Pacing Gates with externally started and stopped resources, be aware that SA z/OS can neither prevent nor delay an externally triggered action against a resource and releases resources as soon as it detects an upcoming external start or stop for them so that the agent does not consider the event to be unexpected.

   Recommended usage is just to put the resources that cause the external actions under the control of a Pacing Gate and to be aware that each one brings along a number of external address spaces when working out your pacing limits.

Monitoring for IPL Completion

A new attribute can be defined in Customization Dialog for entry types Application Group (APG), Application (APL), and Monitor Resource (MTR) to indicate that an IPL is only considered complete if the APGs, APLs and MTRs, defined with this attribute have reached the status of AVAILABLE.

If the attribute 'Monitor for IPL Complete' is set to YES for any APG, APL, MTR resources, additional processing is triggered from the System Automation initialization process. This process checks whether the selected Application Group(s), Application(s) or Monitor Resource(s) have reached the status of AVAILABLE within the specified time limit, and the IPL is indicated as 'successfully completed'. Alternatively, an IPL Complete Status of STANDBY, SOFTDOWN or the compound status SATISFACTORY can be specified in the System Defaults policy.

The attribute may be set to NO or blank, indicating no assessment is made for a resource's availability at IPL completion.

Within policy AUTOMATION OPTIONS of entry type System Defaults (SDF), the time limit is expressed as 'IPL Complete Time Limit' (blank or hh:mm:ss). This attribute allows the specification for the maximum time the resources are allowed to take before reaching the status of AVAILABLE, before the IPL completion as 'over'.

The following messages become available at the console depending on the YES setting(s) you have made for your chosen resource(s):

- ING313I indicates the relevant APG/APL/MTR resources are available and the IPL is complete
- ING314I indicates the relevant APG/APL/MTR resources are not available although specified and the IPL is therefore not complete.
- ING315I indicates that an IPL is complete after the specified time limit, as an optional message.

Refer to the advanced automation variable AOF_AAO_IPL_COMPLETE_MSG in *IBM Z System Automation Customizing and Programming* for additional information on how to customize this function's behavior.

Using Runmodes

The runmode is a flexible way to control the availability of resources without the need to place explicit START or STOP requests against them or manipulating their automation flags.

You can use runmodes for the following purposes:

- Start up your system in two or more stages. You can bring up a subset of your resources, do some checks, bring up another set of resources, do some more checks, and finally bring up the complete system. Note that you can also shut down from an intermediate stage.
- Shut down your system in two or more stages. You can shut down a subset of your resources, apply maintenance, shut down another set of resources, apply some more maintenance, and finally shut down the complete system. Note that you can also bring up your system from an intermediate stage.
- Switch between two or more scenarios with the same configuration. For instance, you can switch between "Weekday" and "Weekend", or "Normal" and "disaster recovery".

To use runmodes, prepare the appropriate definitions in the Customization Dialogs. Valid runmodes must be defined for each system. Each runmode has a name and a of list of so-called "runtokens". Resources like APLs and MTRs also list runtokens. For APGs and system resources, the runtokens are automatically determined by the runtokens and the runtokens of the current runmode. If at least one runtoken is common to both, the resource qualifies for that runmode. A resource also qualifies if it has no runtokens at all and the current runmode contains the special runtoken *NULL. If no runmode is set, all resources qualify as in earlier releases.

A runmode can be set at initialization time as a reply to message AOF603D or by means of the INGRUN command (refer to *IBM Z System Automation Customizing and Programming*). INGRUN additionally allows the forcing of runmode qualification for resources regardless of their runtokens. If such a resource is a group, all members inherit the qualification.

A resource may be qualified for one of the three following reasons:

- Qualification is forced for this resource (see “Setting Runmodes or Forcing a Runmode Qualification using INGRUN” on page 131).
- The resource is a member of a group forced to qualify.
- At least one of its runtokens matches a runtoken designated for the current runmode.

Internally, setting a runmode means injecting a special STOP request against the `sysname/SYG/sysname` resource. This STOP request is generated by the INGRUN command and will therefore be of request method DEFAULT. STOP votes are propagated to all resources of that system. The automation manger then decides according to the above rules whether the STOP vote can be winning or not. All rules for STOP requests apply also to the runmode request. Especially, there can be only one runmode request per system.

Note: When a resource is (forced) qualified for a runmode, it does not necessarily mean that the resource is started. There might be other factors that keep the resource down like the default desired status,

unsatisfied relationships or other existing stop votes. So if you want to start a resource outside of its runmode scope, you may also consider using an appropriate START request.

Runmode requests are persistent as long as you do a HOT start of the automation manger. It means that the request stays in the system until it is explicitly cancelled by setting the runmode to *ALL. The request does not change when the configuration is changed, unless you place a new runmode request. It also means that the decision of the automation manager can be overruled by injecting START or STOP requests with appropriate priority.

To get hands-on experience of runmode, check this tutorial. In this tutorial, you can learn how to define runmode-related policies and how to manipulate the runmodes in runtime to control resource availability.

Defining Runmodes and Runtokens

Runmodes are defined in the customization dialogs using the Entry Type UET (User E-T pairs). The runmode definition includes the runtoken(s) assigned to each runmode.

The entry-type pair INGRUN MODE must be included.

```
Cmd      Keyword                   Data
____     BASIC                     (TBASE1 TBASE2)
____     DISASTER_RECOVERY         (TDISASTER1 TDISASTER2  TDISASTER3)
```

The column **Keyword** specifies the name of the runmode and can be up to 20 alphanumeric characters. Additionally _ and ? is allowed.

The column **Data** specifies the runtokens included in this runmode and must be a list of runtokens, up to 20 characters long consisting of alphanumeric characters ? and _. Additionally a special runtoken *NULL is allowed to qualify all resources which have no runtokens specified.

You can link a User Entry-Type to one or more systems to make runmodes available there.

Note: If you link more than one User Enty-Type containing INGRUN-MODE to a single system, the data is not merged. Instead the last one read takes effect.

Assigning Runtokens

Define runtokens in the customization dialog for entry types APL and MTR within the APPLICATION and MONITOR INFO policy respectively. The **Runtokens** field is available on these policy items.

```
Runtokens.......... TBASIC1 TDISASTER1
```

The **Runtokens** field allows you to specify a list of tokens, up to 20 characters long, consisting of alphanumeric characters, ? and _.

Displaying Valid Runmodes

Display runmodes by using the DISPSYS command for the current system. This section is only displayed if either the current runmode is not *ALL or if there are runmodes defined for the system.

```
    CFG Token    : 20111011163620FF066AAD2094

  Runmode(s)
     Current : TESTPRODWD
     Defined
        TESTBASIC   :  TB
        TESTPRODWD  :  TB WD
        TESTPRODWE  :  TB WE
        TESTPRODREC :  TB WD WE RE

  Message automation table(s)
      DSITBL01 SWTCHTAB INGMSGSA INGMSG01 INGMSGHW

  Flags
     Automation   : Yes
     Init Start   : Yes
```

Figure 29. DISPSYS output for Runmodes

Current shows the runmode currently set. If no runmode is set, it shows *ALL.

Defined shows the list of runmodes defined for the system together with their runtokens.

Note: The complete section "Runmode(s)" is displayed only when runmodes are used.

Displaying Runtokens

Display runtokens for a resource by using the INGINFO command:

```
  Resource   ==>  APPC/APL/AOC5                      format: name/type/system
System     ==> _____________ System name, domain ID or sysplex name

  Schedule             : -None-

  Flags...
     Automation        : YES
     Hold              : NO

  Desired Available    : Always

  Runmode...
     Runtokens     : TB

  Current Order        : -None-
```

```
Runmode...
    Qualifications :  Token, Resource, 2 Groups:   MYGRP1/APG/SYS1  MYGRP2/APG
    Runtokens      :  A1
```

Figure 30. INGINFO output for Runtokens

The **Qualification** field shows how the resource qualifies for the runmode:

- If a **Token** is listed at least one runtoken matches a runtoken of the current runmode.
- If a **Resource** is listed, runmode qualification has been forced for this resource.
- If this resource is qualified because it is a member of groups that are forced to be qualified, the number and names of the groups are displayed.

Runtokens show the runtokens defined for the resource. If the resource is a group, it lists the set union of runtokens of all members.

Note: The complete section **Runmode** is displayed only when runmodes are used.

Displaying Runmode Qualifications

Display runtokens for a resource by using the INGLIST command:

```
 CMD Name        Type System   Category  Subcategory Jobname  Qual
--- ----------- ---- -------- ---------- ---------- -------- ----
    AM          APL  AOC4                          AMA4     --T
    AM          APL  AOC5                          AMA4
    AM_X        APG
    AM2         APL  AOC4                          AMS4     --G
    AM2         APL  AOC5                          AMS5
    AOC4        SYG  AOC4                                   --T
    AOC4        SYS  AOC4                                   --T
    AOC5        SYG  AOC5
    AOC5        SYS  AOC5
    APPC        APL  AOC4                          APPC     --R
    APPC        APL  AOC5                          APPC
    ASCH        APL  AOC4                          ASCH     --T
```

Figure 31. INGLIST output for runmode qualifications

The **Qual** column shows runmode qualification for each resource listed as the following characters:

G
Forced by Group

R
Forced by Resource

T
Runtoken match

-
Attribute does not apply

If runmodes are not used, blanks are displayed.

Displaying Runmode Requests or Votes

Runmode requests or votes for a resource can be displayed via the INGVOTE command:

```
Cmd:  C Cancel request    K Kill request    S Show details    V Show votes
Cmd Name        Type System   Request Data
--- ----------- ---- -------- ------ ----------------------------------------
    AOCA        SYG  AOCA     Req : MakeUnAvailable
                              At  : 2014-10-18  15:46:44
                              Run : Mode=NIGHT_SHIFT Tokens=NIGHTSHIFT1
                                    NIGHTSHIFT2 ...
                              Org : INGRUN(AUTO1)
                              Pri : 01220000  Should Be Down - Default
                              Stat: Pending
    PSVPEVNT    SVP           Req : MakeUnAvailable
                              At  : 2014-10-19  10:00:00
                              Org : SCHEDULE
                              Pri : 01120000  Should Be Down - Schedule
                              Stat: Winning
    PSVPH073    SVP           Req : MakeUnAvailable
```

Figure 32. INGVOTE output for Runmode requests or votes

Select **S** to show more details.

```
Resource. . : AOCA/SYG/AOCA
Request . . : MakeUnAvailable                  Created : 2014-10-18  15:46:44
Originator. : INGRUN(AUTO1)
------------------------------------------------------------------------------
Request priority : 01220000   Should Be Down - Default
Request status   : Pending
STOP type        : NORM
Runmode          : NIGHT_SHIFT
Runtokens        : NIGHTSHIFT1 NIGHTSHIFT2 TOKENALL
Agent Parms      : STATE=AUTODOWN APPLPARMS=**
Restart          : No
Override         : No
```

Figure 33. Detail display for INGVOTE command results

The Runmode and Runtokens fields show the runmode and its runtokens if this is a runmode request.

Note: There can only be one runmode request at a time.

Setting Runmodes at SA z/OS Initialization Time

At SA z/OS initialization time, you can set a runmode as a reply to message AOF603D.

Message AOF767I offers you the following choices.

```
AOF767I AUTOMATION OPTIONS:
   .STOP  -CANCEL AUTOMATION
   .PAUSE  -SUSPEND AUTOMATION
   .NOSTART - DO NOT AUTOMATE SUBSYSTEM STARTUP
   .RUNMODE=x -SET RUNMODE (CURRENTLY *ALL)
   .ENTER  -CONTINUE
```

If you are using runmodes in your configuration you see the runmode currently set. You can reply RUNMODE=x where x is a valid runmode defined on the current system. You can enter RUNMODE=? to see a list of valid runmodes:

```
AOF779I VALID RUNMODES ARE:
  .*ALL
  .BASIC=TBASIC1 TBASIC2
  .DISASTER_RECOVERY=TDISASTER1 TDISASTER2
```

Setting Runmodes or Forcing a Runmode Qualification using INGRUN

Runmodes are also set using the INGRUN command:

```
Request     => SET          Request type  (SET, ADD or DEL)
Target      => ___________  System name, domain ID or sysplex name

--- Parameters for SET request -----------------------------------------
System      => AOCA__       System name
Runmode     => NIGHT_SHIFT___________ Runmode name (mode or ?)
Persistent  => ____         Keep request across IPL (YES/NO)
Type        => _______      Type of processing (NORM/IMMED/FORCE)
Priority    => _______      Priority of request (FORCE/HIGH/LOW)
Override    => _____________________  (ALL/NO/TRG/FLG/DPY/STS/SUS)
Verify      => _____        Check affected resources (YES/NO/WTOR)
Comment     => ____________________________________________________________________
               ____________________________________________________________________

--- Parameters for ADD or DEL requests ---------------------------------
Resource    => _____________________ format: name/type/system
```

Figure 34. INGRUN command dialog (for SET request type)

In order to set a runmode, use the **SET** request option, enter a system name and a valid runmode for the system and press ENTER. The INGREQ verification panel should appear. Pressing PF10 will then inject the runmode request.

Forcing Runmode Qualifications

In order to force a runmode (for a system/sysplex), use the **ADD** request option and a resource name. The resource can be an incomplete specification and it can also contain extended wildcards. You will be presented with a selection panel where you must mark the resource(s) to process and then press PF10 to execute finally the command. This is the case even if only one resource matches your specification. You can see this as a verification panel.

On the selection panel you can enter the line command SEL pattern to select easily any multiple resources. Specify SEL * to select all.

Forced runmode qualification is persistently stored in the automation manager. However, this should be an exception and you should remove unnecessary forced runmode qualifications using INGRUN.

```
Request     => ADD          Request type  (SET, ADD or DEL)
Target      => ___________  System name, domain ID or sysplex name

--- Parameters for SET request -----------------------------------------
System      => __________   System name
Runmode     => _____________________ Runmode name (mode or ?)
Persistent  => ____         Keep request across IPL (YES/NO)
Type        => _______      Type of processing (NORM/IMMED/FORCE)
Priority    => _______     Priority of request (FORCE/HIGH/LOW)
Override    => _____________________ (ALL/NO/TRG/FLG/DPY/STS/SUS)
Verify      => _____        Check affected resources (YES/NO/WTOR)
Comment     => ____________________________________________________________________
               ____________________________________________________________________

--- Parameters for ADD or DEL requests ---------------------------------
Resource    => MYAPL1/APL/SYS1___________ format: name/type/system
```

Figure 35. INGRUN command dialog (for ADD request type)

In order to delete the forced runmode qualification specify request **DEL** and a resource name. The resource name can be an incomplete specification and it can also contain extended wildcards. You will be presented with a selection panel as for the ADD request. You also specify a question mark (?) which lists all resources previously processed by an ADD request.

Using UP Status Delay

SA z/OS recognizes that applications reach an UP status by trapping a specific message that indicates the application is running.

There are some applications that do not issue an appropriately-timed message. In these cases, SA z/OS is instructed to trigger the resource's transition to UP from a message that is issued earlier in the start sequence. However, in some situations there can be problems when commands for the dependent applications are sent while the parent application is still initializing.

The UP Status Delay setting in the application's APPLICATION INFO policy adds a pause between the processing of the message that indicates that the resource is up and the transition of the resource into an UP status. The effect is to delay the processing of the dependent resources, giving the application time to finish initializing.

Using Schedules

Using service periods (also called schedules) is one of the methods to set goals for resources. Service periods are defined as permanent automation policy in the customization dialog.

To display all the service periods that are defined in the sysplex, use the INGSCHED command dialog.

On the INGSCHED panel, you can enter the following command codes for a selected schedule:

A
Using this action code to view the From-To time windows of the selected service period for the week. No updates are possible on this panel.

B
Displays the actual From-To time windows for the selected schedule including the overrides, starting from the requested day (initially it is the current day). Use this option if you want to override a schedule for all resources that use this schedule (that is, for all resources that this schedule is linked to in the customization dialog). Such an override is called a *schedule override* and is described in "Schedule Override" on page 133.

C
Displays the resources that are using the selected schedule. You can select a resource to view the schedule that is used for this resource. You can override the schedule just for the selected resource. Such an override is called a *resource override* and is described in "Resource Override" on page 133.

The column with the **UP** and **DN** entries indicates whether the schedule is an UP or DOWN time window.

Schedule Override

You can interactively override a schedule for all resources that use this schedule. You can interactively override the schedule starting from a specified starting date for selected dates in the near or distant future.

To reach the schedule override panel, use action code **B** against a schedule on the INGSCHED panel, or enter the INGSCHED command with the schedule name as a parameter. Schedules that have been overwritten are shown in yellow. From this panel you can overtype or complete the displayed schedule.

Resource Override

You can interactively override a schedule for a particular resource that uses this schedule. You can interactively override the schedule starting from a specified starting date for selected dates in the near or distant future.

There are three ways to reach the resource override panel:

- Use action code **C** against a schedule on the INGSCHED panel and then, from the resource list that is displayed, select the resource that you want to override the schedule for.
- Issue the INGSCHED command with the resource name as a parameter, for example:

```
INGSCHED AGFG2/APL/AOCA
```

- From the INGLIST command dialog (see “Using INGLIST to View Resources” on page 115), issue action code **I** against a resource.

Service windows that are overrides for the resource are shown in pink. You can edit the schedule by overtyping new values.

From the INGSCHED panel, which is specific for the current resource, you can:

- Enter new service times. Pressing Enter displays the schedule override in pink.
- Overtype or blank out existing service times. Pressing Enter displays the schedule override in pink.
- Delete a complete schedule override on a particular date (displayed in pink) by entering action code **D** in the command field. The schedule override is removed and the general schedule that has been defined for the current resource is restored and displayed in green.

Displaying and Setting Trigger Information

Triggers are a very powerful tool of goal driven automation. Because they influence the behavior of applications and application groups, the automation administrator needs to be very careful when connecting triggers to resources.

For information on event and trigger support refer to *IBM Z System Automation Defining Automation Policy*.

In SA z/OS two commands are available to display trigger information:

INGTRIG
This command shows all triggers that are defined and used by resources within the automation manager scope. You can view all resources that use a trigger by entering the action code **S** in its command field. Entering the action code **S** against a resource on this panel shows the trigger details for the resource (that is, it invokes the DISPTRG command for the resource).

DISPTRG
This command displays the trigger conditions for a resource. To display the DISPTRG command dialog for a resource, enter, for example:

```
DISPTRG BATB000P100
```

If you enter this command without parameters or SA z/OS cannot find the resource that you specified, SA z/OS prompts you for more information to help locate the resource. If there is more than one resource that matches the resource specification, you are presented with a selection list.

Use command code **S** against a trigger to view details about the events that must be set to activate it. This displays the Trigger Condition List. From this panel you can use the following command codes against an event:

- **S** to set or **U** to unset the events that have been defined for the resource
- **D** to view a list of the resources that use the event as a trigger

Displaying and Setting Events

You can display and set event information at different levels in your enterprise:

- DISPEVT shows resource information for a particular event.
- DISPEVTS shows all events in a sysplex.
- INGEVENT allows you to set or unset an event for a particular resource or all resources on one system, or all systems in the sysplex.

DISPEVT
You can reach the DISPEVT command dialog by issuing the DISPEVT *event_name* command or using the command code **D** from the DISPTRG trigger condition list.

This panel shows all resources that have the specified event in their trigger definition. The following information is displayed for each resource:

- The name of the resource
- The resource type
- The name of the system where the resource resides
- The name of the trigger that the resource is linked to
- The status of the event (SET or UNSET). If the **Status** column is blank this means that the status is unknown. This is treated as UNSET.

DISPEVTS
You can reach the DISPEVTS command dialog by issuing the DISPEVTS command from any command line. This displays the events that are defined for resources in the sysplex and the condition when the event is reset for the resource.

From this command dialog, you can issue the following action codes against an event:

- **D** to show related resources (that is, it invokes DISPEVT for the event)
- **S** to set or unset the event (that is, it invokes INGEVENT for the event)

INGEVENT
With the INGEVENT command, you can set or unset an event for a specified resource or for all resources in the sysplex that it is defined for.

To reach the INGEVENT command dialog issue the action code **S** against an event in the DISPEVTS command dialog.

Note: With the line mode INGEVENT command, you can directly set or unset an event for all resources or a specified resource, for example:

```
INGEVENT BATCHSTP REQ=UNSET RESOURCE=BATB000P200/APL/AOC7
```

Successful completion produces the following message:

```
AOF442I UNSET EVENT BATCHSTP DONE FOR RESOURCE BATB000P200/APL/AOC7
ON AOC7. - FROM=AOC7
AOF099I FUNCTION COMPLETED
```

Determining Automation Agent Flag Values

Automation flags can be predefined in the customization dialogs or can be set during run time.

Their meanings are explained in detail in “SA z/OS Automation Flags” on page 61.

To determine the actual flag values that have been set in the customization dialog or during runtime, together with the effective flag values for a particular subsystem, use action code A to call DISPFLGS for this subsystem. It displays the actual and effective flag values for the selected subsystem and for related minor resources.

Each flag has one of the following values:

–
There is no explicit setting.

Y
The flag is turned on.

N
The flag is turned off.

E
The value of the automation flag depends upon the values returned by one or more user exit and will be determined when the value is required.

L
Resulting commands or replies are logged only.

S
The flag is turned off, because the resource is suspended.

Displaying Actual and Effective Automation Agent Flags

Enter DISPFLGS at any command line to display the actual flags that have been set in the customization dialog or during runtime, and the effective flags for any resources that do not have explicit flags coded for them.

The DISPFLGS command dialog displays the following:

- The name of the system where the resource resides.
- The resource name. Two types of resources are displayed:
  - Major resources are subsystems or generic settings (for example, SUBSYSTEM).
  - Minor resources are specific situations or resources within a major resource. Minor resources are indented one space to the right under their major resource.
- The flags that have been set for the resource.
- The flags that are in effect for the resource.

If a resource is suspended, no command codes are possible for the subsystem or resource.

If a resource is not suspended, the following command codes are supported:

A
Allows you to define automation flags for a resource using the same flag settings as the selected resource. Another panel is displayed where you can specify the resource name and optionally overtype the flag settings.

C
Allows you to modify the automation flags of the selected resource. Another panel is displayed showing the current flag settings where you can overtype the flag values.

R
Causes to reset the automation flags to the values specified in the automation configuration file (ACF).

S
Shows the scheduled override settings of the automation flags for the selected resource. You can specify scheduled overrides using the customization dialog. These are specific times when the automation is turned off for a particular flag and resource.

Displaying Automation Agent Flag Override Settings

The customization dialog allows you to turn automation flags off for a scheduled time period. You can view these time period settings by issuing the DISPSCHD command, which leads to the DISPSCHD command dialog panel.

- The **System** field shows the name of the system where the resource is defined.
- The **Resource** field shows the name of the resource.
- The **Fl** (flag) field shows an abbreviation of the automation flag name.
- The **Day** field shows the day of the week that automation is turned off.
- The **Start** and **End** fields show the time periods that automation is turned off.

Setting Automation On or Off: INGAUTO

You can change the automation setting for a specific automated resource or a group of automated resources using the INGAUTO command dialog. You can turn all automation off or only particular automation flags.

Note:

- Be sure to specify the correct subcomponent name. Although the subcomponent does not exist, the message `FUNCTION COMPLETED` will appear.
- For suspended resources, it is not possible to change the automation settings via INGAUTO.

Enter INGAUTO on any command line to reach the INGAUTO command dialog panel.

In the **Resource name** field, type the name of the resource that you want to determine automation flags for. If you specify the value `defaults`, you set automation flags for all resources that do not have more specific automation flags defined or that do not belong to a group that has flags defined.

You can use the **Interval** field to turn automation on or off for all or selected flags for a specific time period. In this field, type the amount of time you want the automation turned on or off. This time period begins immediately when you press the Enter key.

At the end of the interval, the flag (or flags) will be reset to the value defined in your automation control file. If you leave the **Interval** field blank, the flag (or flags) will remain changed until they are manually reset or updated.

Working with Application Groups

An application group is a resource that both summarizes the status of other resources (its members) and controls the activation of those members.

A special feature is that the members can inherit the group's dependencies.

Each member is a resource, as is the group itself. This means that groups can contain other groups. You can issue commands to start or stop groups as an entity. Also, you can link a group to a service period and thus schedule it automatically.

Detailed information about how to organize applications into a group, nest groups into other groups, and generate resources from groups and applications is provided in *IBM Z System Automation Defining Automation Policy*.

The concept of resources allows you to dynamically move applications and groups from one system to another system in the sysplex. This move is achieved by stopping an active set of resources and starting an alternate set in a coordinated fashion. With this concept, you can also start and stop equivalent resources to maintain a desired level of server capacity.

There are two types of application groups:

- **System Application Groups**: Application groups that are associated with particular systems.
- **Sysplex Application Groups**: Application groups that are associated with a certain sysplex. The application group members can be spread over multiple systems within this sysplex.

Groups can also have one of three different *natures* (see Figure 36 on page 138):

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 that will be started if the 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. With SERVER application groups you can specify what should happen if their components fail to start.

The automation administrator can set the numbers of members that should be available for a SERVER application group by specifying the *Availability Target* when defining the group.

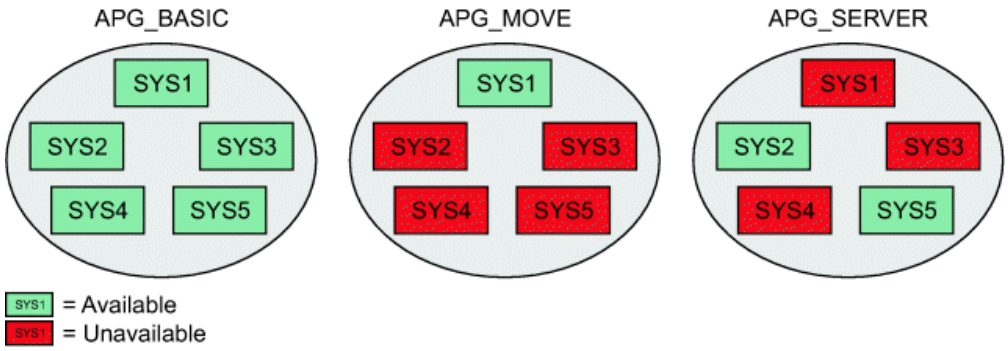


Figure 36. Natures of Application Groups

Preference values are used for SERVER and MOVE groups to determine which members should preferably be made available.

For detailed information on how to define application groups and what nature to use for different purposes, refer to *IBM Z System Automation Defining Automation Policy*.

You can use the DISPAPG command to display detailed information about an application group from the automation agent point of view, such as automation agent or automation manager related information or automation policy information.

Using INGGROUP

This section introduces the functions that are provided by the INGGROUP, gives details of application move, and describes the options you can use with the INGGROUP command dialog.

Functions

The functions provided by the INGGROUP command are:

- Automatic group management to maintain application availability. This includes the activation of backup or alternate resources in response to resource or system failure.
- A superset of the behavior provided by Automatic Restart Manager. Although INGGROUP does not provide a true cross-system restart, it is able to take recovery actions in a variety of situations where Automatic Restart Manager is not. These include application failure, supporting application failure, and 'total active instance' based management (for SERVER groups).
- The ability to move resources away from a system, on either a group or subgroup level or a system level.
- The ability to query and dynamically change a group's policy to stop one instance of a resource and start another.
- The ability to view an estimate of the impact of those changes before committing them.
- The ability to perform actions resulting from the above changes in a delayed, non-disruptive fashion.
- The ability to return resources to their normal systems in a delayed, non-disruptive fashion.

Application Move

Thus one of the most important functions of the INGGROUP command is moving resources around in a sysplex.

This means being able to move applications or application groups from one system to another, that is, stopping an active set of applications (dependent and supporting resources) on one system and starting them on another system without loss of information. This can be advisable for the following reasons:

- To ensure availability of resources throughout a sysplex despite single system failure
- To balance workload in a sysplex
- To enable complete applications to be moved away from a certain system if this is needed for other work or maintenance

Application move is implemented in SA z/OS in two flavors:

- A planned or scheduled move, which is defined by automation policy and where the availability of resources is automatically ensured through defined availability targets and preference values for the single resources. For detailed information, refer to *IBM Z System Automation Defining Automation Policy*.
- An immediate move for emergency reasons. To do this, you can issue the INGGROUP command against a group to move it.

When an application has been automatically moved to a backup system after a problem, the policy can be set to return it to its home system either as soon as the home becomes available (causing an application outage) or at the next (scheduled) application outage.

For more information about application move see "Controlling Application Move and Server Management" on page 52.

INGGROUP Options

You can invoke the INGGROUP command dialog either by typing the command name at the command line or by selecting option **J** from the INGLIST command dialog.

INGGROUP allows you to change the policy of application groups. You may specify:

- The target sysplex for the command via the **System** field. If omitted, this defaults to the local system/sysplex. Specify the system name only when viewing or changing the policy of an application group that is not part of the local sysplex.
- The policy change that you want to make through the **Action** field.
- Next you must specify either the name of the target group or a set of templates that will match one or more groups. All policy items work with a list of group name templates and apply to all Move and Server groups matching those names. While applying a policy to a Basic group is not an error, it will not achieve anything either.
- For the EXCLUDE-AVOID-INCLUDE option, you can specify a list of systems that the policy is to be set for. Specify a question mark (?) at the **System(s)** line to see a list of systems within the sysplex.

  This option indicates that you want to remove movable members from a system (exclude), prohibit SA z/OS from activating movable members on a system (avoid) or that you wish to undo either of the other two actions (include).

  Note: The AVOID/EXCLUDE options are only applied to the members of the group that are applications. They are not applied to members of a group that are also groups. In other words, these options are not recursively spread out to nested application groups.
- With the PACIFY option, you can make the specified group passive. This option will immediately revoke all votes from all of the group members. This may cause the desired state of some of its members to change, resulting in them being started or stopped.

  With the ACTIVATE option, you can make a group active. When a group becomes active, SA z/OS will propagate *MakeAvailable* requests to the group members that are required to make the group available. Those members that are not required will receive a *MakeUnavailable* request. Note that if a member has

a preference of 1 it will always be propagated a *MakeUnavailable* request and if it has a preference of 0 (a passive member) it will never be propagated any sort of request.

- With the ADJUST option you can indicate that you want to temporarily adjust a group's policy.
- With the RESET option you can set the preference value assigned to each member of the group to its initial value.
- With the DEFAULT option you can set the availability target, satisfactory target, group passive attribute and the member's preference value to the value defined in the policy.
- With the OVERRIDES option you can display any overrides that have been made with the INGGROUP command for the specified resource groups.
- With the POLICY option you can display the policies for the specified resource groups.
- With the RECYCLE-CANCEL option you can initiate or halt a rolling recycle of the specified groups. This affects Server and Move groups that are within the scope of the command. Use the CHUNK parameter to recycle group members in parallel, improving performance.

  Note that if the systems in a Server group all have the same preference value, the system that is active after a rolling recycle has completed may not be the same one as before the rolling recycle. Thus, triggering a rolling recycle always results in a move occurring. If you set the preference of the original system more than 250 points higher than that of any other system, the rolling recycle is followed by a second move to return the application to that system. This leads to an increase in the application down time.

Moving Sysplex Application Groups

INGMOVE makes moving sysplex application groups easier. Rather than manipulating the preference value of each member in the sysplex application group, you simply specify where the group should be moved to.

In a sysplex application group of type MOVE (referred to as a *move group*) only one member is active at a time. By specifying the new location of the move group, the active member is terminated and the member associated with the new location is activated.

If you enter INGMOVE at the command line, the INGMOVE command dialog displays all the active move groups in the sysplex. You can specify the name of one or more sysplex application groups, separated by a blank or a comma, and each name can contain a wildcard, for example, `ingmove movplex*`. The panel then displays all sysplex application groups of type MOVE that match the specified filter criteria.

You can also specify a WAIT period, which is the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds and, if it is not specified, it defaults to 30 seconds.

On the INGMOVE panel, the **Group name** column shows the name of the move group. Because the move groups are unique within a sysplex, only the first part of the resource group name is shown.

The **Obs Status** column shows the observed status of the move group as seen by the automation manager.

The **Systems** columns show a list of the systems where a member of the MOVE group exists and could run. Different colors are used to indicate the state of the system or the member of the group that is running on that system, as follows:

- Red if the system is down
- Yellow if the member of the group that is running on that system is not startable
- Green if a member of the group is currently active on that system
- Pink if the system is excluded
- White if the system is avoided
- Turquoise in all other cases

Systems that are underlined have the highest base preference value and are called the home system (or systems).

The system that has the highest actual preference is shown in reverse video, unless it is a home system. Note that this can be more than one system.

The **Move to** column is an input field where you can specify the system that the group should be moved to. Moving a sysplex application move group means terminating the member that is currently active and starting the member on the designated system. You can designate the system by specifying:

- The name of the system in the input field.
- An asterisk (*). This means that the group is moved back to its home system. If more than one home system exists, it is moved to the 1st one in the list.
- An equals sign (=). This is a shorthand that means to take the value from the nearest field above with a value specified in it.

Setting Timers

You can set a timer to issue commands at a specific time or interval of time for your convenience. Therefore, you do not need to be present at a console to perform a repetitive task at a certain time.

This NetView timer is actually the SETTIMER command that issues a command or list of commands at a specified time or specified interval.

Invoking the SETTIMER command displays a list of existing timers. From this panel, you can perform the following tasks.

Table 16. Setting Timers

Task	Action on the SETTIMER command dialog panel
Create a timer	Type add at the command line and press Enter. On the Add/Change Timer panel that appears fill in the fields as required. **Note:** If you want to add a new timer but use most of the same values as a timer that has already been defined, type A in the **CMD** column and modify the SETTIMER panel.
Modify a timer	Enter C in the **CMD** column for the timer that you want to change. The Add/Change Timer panel is then displayed where you can modify the settings of your selected timer.
Delete a timer	Type D in the **CMD** column for the timer you want to delete and press Enter.
Suspend a timer	Enter U in the **CMD** column for the timer that you want to suspend. **Note:** The timer is deleted from NetView but all its relevant data is kept in common global variables. Only CHRON timers can be suspended.
Resume a timer	Enter R in the **CMD** column for the suspended timer that you want to resume (or reactivate).

Interacting with the Automation Manager

You can operate the automation manager itself with the INGAMS command to:

- Display information about all currently registered automation managers and automation agents
- Change the operation mode of an automation manager from secondary to primary
- Refresh the configuration data used by the automation manager
- Perform diagnostic functions

The INGAMS command displays all currently registered automation managers and automation agents. The **Role** column describes what role the member has, as follows:

AGENT
An automation agent

PAM
A primary automation manager

SAM
A secondary automation manager

Changing the Automation Agent Status

You may want to change the status of an application in the following circumstances:

- When you want SA z/OS to resume handling of an application after an event that causes an application to go to a non-automatable status (for example, STOPPED, CTLDOWN, BROKEN, ACTIVE).
- When you want to prevent automation from restarting an application by setting its state to CTLDOWN.
- When you want to inform SA z/OS that an application that has been shut down is going to be moved to another system.

System operations defines 24 different statuses for automation, described in “Automation Agent Statuses” on page 70. You can change the automation statuses with the SETSTATE command (described in *IBM Z System Automation Operator's Commands*).

The SETSTATE command is used to change the automation status of a subsystem or a group of subsystems (supporting and dependent resources). The status change you are able to make depends upon the subsystem's application monitor status and whether it is suspended or not:

- If the subsystem is suspended, it is not possible to change its automation status. At first, it has to be resumed.
- If the subsystem's application monitor status is ACTIVE, you can change its automation status to UP.
- If its application monitor status is INACTIVE, you can change it to RESTART or CTLDOWN. You may also be able to change it to MOVED or FALLBACK, depending upon its system associations.

With the SETSTATE command dialog, you can change the automation agent status of a subsystem or a group of subsystems with the **Scope** field.

Monitoring of Resources

With SA z/OS you can monitor various aspects of your resources, including:

- Messages, “Monitoring Messages” on page 142
- Hardware, see “Monitoring Hardware” on page 145
- Health performance, see “Monitoring Health Performance” on page 146
- OMEGAMON sessions, see “Monitoring OMEGAMON Sessions” on page 147
- Processor status, see “Monitoring Processor Status” on page 150

Monitoring Messages

Message suppression facilities at the target system determine the amount of message traffic sent from a target system sent to a console. Messages that do not normally require operator intervention are usually suppressed at the target system. Messages that do normally require operator intervention may also be suppressed at the target system as automation programming is developed to respond for you.

You can monitor messages at the SA z/OS workstation via several mechanisms and at several levels. You can view:

- No messages

- Exceptional messages
- Messages that have been sent to a specific system console or operator console
- Messages that are related to the control of processor operations-managed resources
- SA z/OS messages (specified by message class)

How to Display Outstanding WTOR Messages

WTOR (Write to Operator with Reply) messages are z/OS messages that require operator action. These messages can be viewed on the z/OS operator console.

How to Display Exceptional Messages

An exceptional message is a message whose severity is either Unusual, Important or Critical. You use the INGMSGS command to display exceptional messages. From the INGMSGS panel you can either delete the messages or view details about the resource that is associated with the message (that is, by invoking INGINFO for it).

By default the messages are displayed on the INGMSGS panel in chronological order (the oldest message is shown at the top of the panel) but can be rearranged in any other order, for example, by severity with the SORT subcommand. The individual messages are colored depending on their severity. The color attribute is defined via SDF status definitions.

You can also display exceptional messages for one or more resources by specifying them with the INGMSGS command.

For more details about the INGMSGS command, see *IBM Z System Automation Operator's Commands*.

How to Display Messages for a Specific System or Operator Console

You can choose to receive all messages for a specific target system operator console using the Processor Operations Interested Operator List. These messages are displayed at your NetView console session.

Messages from the target console that are displayed include new message lines displayed on the Support Element console integration (CI) window, sent by z/OS, OS/390, VM, Linux on System z, VSE, or the Coupling Facility Control Code (CFCC). Multiline messages issued by z/OS look like several single-line messages from the console. Similarly, if an z/OS message extends over two lines, it looks like multiple separate messages from the processor operations console.

When you are monitoring a CI, make sure it is enabled to receive regular messages. For z/OS, refer to the VARY CONSOLE command parameters ACTIVE, DEACTIVATE.

All processor operations messages from a target system begin with the processor operations identifier ISQ900I or ISQ901I plus the target system name and the console type. Messages prefixed by ISQ900I are not displayed but are used by the automation. User automation should key on the ISQ900I messages. Messages prefixed by ISQ901I are sent to interested operators and should not be used for user automation. Multiline messages appear as multiple messages.

Joining or Leaving an Interested Operator List

For target system operator console messages from CI use the ISQXMON command. For general processor operations resource control information messages use the ISQCMON command on a NetView console.

Viewing the List of Interested Operators

The processor operations status facility contains status panels showing the list of interested operators for a specific target system.

You cannot directly access the Processor Operations Interested Operator List panel with the ISQXDST command. You can access the Processor Operations Interested Operator List panel only from the following status panels:

- Target System Summary panel.
- Target Hardware Summary panel. This is accessible only from other status panels.

The best way to access the Processor Operations Interested Operator List panel is to issue the following command to access the Target System Summary panel for a specific target system:

```
ISQXDST target_system_name
```

The resulting status panel shows the communication path status for the connection between the processor operations focal point and the target hardware (CPC) where the target system image is running (LPAR or Basic Mode). Place the cursor on the communication status line and press the PF7 key to see the associated Processor Operations Interested Operator List panel. If your operator identifier (the name you use to log on to NetView) is in this list, you should be receiving messages from that path. All other operators whose names are in the list also receive messages from that path.

You check the interested operator list for processor operations resource control messages (the ISQCMON list) by issuing the following NetView command:

```
LIST ASSIGN=GROUP
```

The interested operator list for processor operations is in group +TSCFMON.

How to See SA z/OS Messages Specified by Message Class

Notification operators are defined during customization of SA z/OS using the SA z/OS customization dialog.

A notification operator is a NetView console operator who is authorized to receive messages from SA z/OS. The classes of messages that a notification operator is authorized to receive are specified at the time a notification operator is defined. See the tasks below to understand how to display information about a specific notification operator, how to add and delete notification operators, and how to change the messages that a notification operator receives.

Automation manager messages can be received automatically.

Also, you can receive messages from all systems within the sysplex.

How to Get Notify Messages: INGNTFY

This section provides information about notification operators and those messages that notification operators should be notified of.

Display Settings for a Notification Operator

If you specify the INGNTFY command without a parameter, the main INGNTFY panel is displayed, showing a list of all the notification operators that have been defined. From this panel you can:

- Add (*A*) a notify operator using the settings of the selected operator
- Show or change settings (*C*) for the selected operator
- Delete (*D*) the notify operator
- Turn off (*O*) the sending of notify messages to the selected operator

For further information on the INGNTFY command refer to the *IBM Z System Automation Customizing and Programming*.

Query a Specific Notification Operator

To display information about a specific notification operator, type INGNTFY on any command line and type `query` into the **Status/Action** field. Also, type the ID of the operator you want information about.

Turn On or Off all Notification Messages

Use INGNTFY OFF on a NetView or operator console to temporarily turn off messages when you do not want to see them. Use INGNTFY ON on a NetView or operator console to turn the messages back on when you want to see them again.

Note: Each Notify Operator is issued a copy of the originating message. Consequently, if this message is being automated (it has an entry in the NetView automation table) this automation is replicated for each Notify Operator.

Add a Notification Operator

You can add more notification operators through the INGNTFY command dialog. Remember that this operator will only be able to receive messages until the next time the automation control file is loaded, when all changes made using the operator interface are lost.

To add a notification operator, enter INGNTFY on any command line. Type add in the **Status/Action** field. Also, type the ID of the operator you want to add.

In the **Classes** field, type the classes of messages you want this operator to see. The default classes are 40 and 80.

If you want to hold certain types of messages on this operator's screen, move the cursor beside the desired Message type (Information, Eventual Action, Immediate Decision, System Wait, or Immediate Action), and type any character. For more information on message types, see *IBM Z System Automation Messages and Codes*.

Change the Messages a Notification Operator Receives

To change the messages you or another operator receives, you access the INGNTFY command dialog and you fill in the panel exactly as if you were adding a new operator. (You do not have to delete the definition first because the new definition overwrites the old.)

Because the new definition overwrites the old, you need to be careful when you are changing the messages an operator receives. For example, if the operator was originally assigned to receive classes 41 and 43, and you typed 44 in the Classes field, 44 would be the only class that the operator would now receive. If you want to add a class, you must also type all the classes that were previously assigned.

You can perform a query to find out what messages the operator currently has assigned. See "Query a Specific Notification Operator" on page 144 for more information.

Delete a Notification Operator

You can also use the INGNTFY command dialog to delete notification operators. Remember that if you use this panel to delete an operator, the deletion will be in effect only until the automation control file is reloaded.

To delete a notification operator, type `delete` in the **Status/Action** field. Also, type the ID of the operator you want to delete. This field can contain any valid NetView operator or the keyword SYSOP. If SYSOP is defined as a notification operator its function is to receive messages to the master console when no other operator is logged on, or to receive specific messages depending on the message classes assigned.

Monitoring Hardware

You can monitor all processors using processor operations.

General Information About Panels: Pressing the Enter key toggles between a dynamically updated panel and a static panel. If any processor operations screen is updated frequently, it appears to flicker. You may need to place the panel into a static condition to select another function. The **Updates** field in the top right of the panel shows the current mode of the panel (Dynamic or Static).

Processor Operations Status Information

Processor operations allows you to monitor target processors, target systems, and focal point communication path by using the ISQXDST status panels of your NetView operator console.

These panels are available only on a NetView operator console that is attached to the SA z/OS processor operations focal point system.

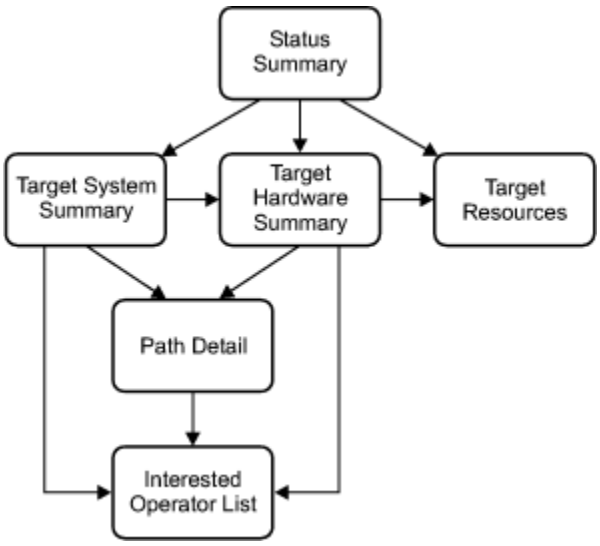


Figure 37. ISQXDST Command on NetView Console

Displaying Processor Resource Information (SNMP Connections)

To view target hardware summary of a processor, select PF9 from the Target System Summary panel or from the Status Summary panel on a NetView console.

This panel has the following PF keys:

- The PF7 key displays the Processor Operations Interested Operator List panel (also accessed with the ISQXMON command).
- The PF11 key displays the Path Detail panel, which provides detailed status information about a specific NetView path. First place the cursor on the line with the name of the path that you want more information about, and then press PF11.

The Target Hardware Summary panel provides detailed status information about the target hardware that the target system is defined on, including:

Initialized Target Systems
(Updated dynamically) List of target systems currently initialized to this target hardware.

Communication Path Error Status
(Updated dynamically) Status information in case of a communication problem between the NetView focal point and the target hardware.

Monitoring Health Performance

You can monitor the health performance of your resources using monitor resources with:

- The Network Communications Control Facility (NCCF) panels, see "Using the NCCF Panels" on page 146
- The Status Display Facility (SDF), see "Monitoring with the Status Display Facility" on page 153

Using the NCCF Panels

Once you have loaded your configuration you can use the NCCF panels to check the status of your resources. You might want to start with INGLIST.

If you scroll to the right using the PF11 key, you can see the **Health** status column.

Suppose that you are monitoring CLIENT/APL with INGLIST and you find a compound status of DEGRADED and a health status of WARNING. In this case you can enter the command code M in the **CMD** entry field of CLIENT/APL. This displays the DISPMTR panel showing all monitor resources (MTRs) that are connected to CLIENT/APL with HasMonitor relationships. To find out the reason for the health

status WARNING you can scroll to the right to find more information or enter the command code D in the **CMD** entry field of the MTR, which takes you to the DISPMTR Details panel.

Monitoring OMEGAMON Sessions

SA z/OS allows you to use Monitor Resources to connect to classic OMEGAMON monitors to send commands and receive responses.

You can use the INGSESS command either from the command line or with the fullscreen command dialog to:

- Display OMEGAMON sessions
- Start sessions manually to test connection and authorization
- Stop sessions to perform maintenance
- Show additional session attributes, for example, logon data, timeout, statistics, etc.

Displaying OMEGAMON Sessions

When you enter the INGSESS command at the NetView command line the INGSESS command dialog is displayed.

The panel displays the following session information:

- The **Session** column shows the name of the session that represents an OMEGAMON monitor.
- The **System** column shows the system that established the connection to OMEGAMON.
- The **Type** column shows the type of session, that is, the OS or middleware that the OMEGAMON monitor is monitoring.
- The **Status** column shows the status of the session. It can be one of the following:

  INACTIVE
  The session does not exist

  ACTIVE
  The session exists and is ready to receive requests

  AUTHFAIL
  The session could not be created because the logon to OMEGAMON failed; all requests will be denied

  SESSFAIL
  SA z/OS received unexpected data or some NetView communication error occurred; all requests will be denied

  MAINT
  The session was terminated for maintenance purposes; all requests will be denied
- The **Appl-id** column shows the name of the OMEGAMON VTAM application as defined by the installation during customization of the OMEGAMON product.
- The **User id** column shows the user that is defined to log on to the OMEGAMON application. The user ID is needed to control access to the OMEGAMON application. It may be blank if product level security is not implemented for this OMEGAMON application.
- The **SessOper** column shows the automated function name that was assigned to the session by SA z/OS during initialization.

Starting OMEGAMON Sessions

You can start sessions manually to test connection and authorization using either:

- The command line, for example

```
INGSESS ims742cr req=start outmode=line
```

- The command dialog, as follows:

```
CMD:  B Start session   C Stop session   D Details

CMD Session     System   Type     Status     Appl-id  User id  SessOper
--- ----------- -------- -------- ---------- -------- -------- --------
 _  CICSKY41    OMIICICS AOC9     ACTIVE     IPSPOC0  SAOM     AOFSES01
 _  DB2SGG4     OMIIDB2  AOC9     INACTIVE   IPSPD2C  SAOM     AOFSES02
 _  DB2SG14     OMIIDB2  AOC9     MAINT      IPSPD2C  SAOM     AOFSES03
 B  IMS742CR    OMIIIMS  AOC9     INACTIVE   IPSPOI0  SAOM     AOFSES01
 _  OMSY4MVS    OMIIMVS  AOC9     AUTHFAIL   IPSPM2RC SAOM     AOFSES02
```

This creates a new session with the appropriate OMEGAMON, if it does not already exist (that is, the session status is INACTIVE or MAINT). After successful session creation, the session status is ACTIVE. The session remains active until it is stopped, either explicitly (using command code C), or implicitly by NetView on behalf of OMEGAMON or session task termination.

If the session could not be started, the status will become either AUTHFAIL or SESSFAIL.

Stopping OMEGAMON Sessions

You can stop sessions to perform maintenance using either:

- The command line, for example:

```
INGSESS cicsky41 req=stop outmode=line
```

- The command dialog, as follows:

```
 CMD:  B Start session   C Stop session   D
Details


 CMD Session     System   Type     Status     Appl-id  User id
SessOper
 --- ----------- -------- -------- ---------- -------- --------
--------
  C  CICSKY41    OMIICICS AOC9     ACTIVE     IPSPOC0  SAOM
AOFSES01
   _  DB2SGG4     OMIIDB2  AOC9     INACTIVE   IPSPD2C  SAOM
AOFSES02
   _  DB2SG14     OMIIDB2  AOC9     MAINT      IPSPD2C  SAOM
AOFSES03
   _  IMS742CR    OMIIIMS  AOC9     INACTIVE   IPSPOI0  SAOM
AOFSES01
   _  OMSY4MVS    OMIIMVS  AOC9     AUTHFAIL   IPSPM2RC SAOM
AOFSES02
```

This destroys the session with the appropriate OMEGAMON. The session status will be changed to MAINT. The session is disabled to prevent it from being created implicitly again.

After you have fixed the problem you can restart the session.

Recovering from Session Failure

A session may not be created because either the logon to OMEGAMON failed, or SA z/OS received unexpected data or some NetView communication error occurred.

About this task

In these cases all requests to the session will be denied.

To recover a failed session:

Procedure

1. Put session in maintenance mode, for example:
   - From the command line:

     ```
     INGSESS omsy4mvs req=stop outmode=line
     ```

   - Using the INGSESS command dialog:

     ```
      CMD:  B Start session   C Stop session   D
     Details


      CMD Session     System   Type     Status     Appl-id  User id
     SessOper
      --- ----------- -------- -------- ---------- -------- --------
     --------
        _  CICSKY41    OMIICICS AOC9     ACTIVE     IPSPOC0  SAOM
     AOFSES01
        _  DB2SGG4     OMIIDB2  AOC9     INACTIVE   IPSPD2C  SAOM
     AOFSES02
        _  DB2SG14     OMIIDB2  AOC9     MAINT      IPSPD2C  SAOM
     AOFSES03
        _  IMS742CR    OMIIIMS  AOC9     INACTIVE   IPSPOI0  SAOM
     AOFSES01
       C  OMSY4MVS    OMIIMVS  AOC9     AUTHFAIL   IPSPM2RC SAOM
     AOFSES02
     ```

2. Analyze and fix the problem
3. Restart the session, for example:
   - From the command line:

     ```
     INGSESS omsy4mvs req=start outmode=line
     ```

   - Using the INGSESS command dialog:

     ```
      CMD:  B Start session   C Stop session   D
     Details


      CMD Session     System   Type     Status     Appl-id  User id
     SessOper
      --- ----------- -------- -------- ---------- -------- --------
     --------
        _  CICSKY41    OMIICICS AOC9     ACTIVE     IPSPOC0  SAOM
     AOFSES01
        _  DB2SGG4     OMIIDB2  AOC9     INACTIVE   IPSPD2C  SAOM
     AOFSES02
        _  DB2SG14     OMIIDB2  AOC9     MAINT      IPSPD2C  SAOM
     AOFSES03
        _  IMS742CR    OMIIIMS  AOC9     INACTIVE   IPSPOI0  SAOM
     AOFSES01
       B  OMSY4MVS    OMIIMVS  AOC9     MAINT      IPSPM2RC SAOM
     AOFSES02
     ```

Displaying Additional Information

You can display additional session attributes (for example, logon data, timeout, statistics, etc.) using either:

- The command line, for example:

```
INGSESS omsy4mvs req=detail outmode=line
```

- The command dialog by entering the action code D for a session, as follows:

```
CMD:  B Start session   C Stop session   D Details

CMD Session     System   Type     Status     Appl-id  User id  SessOper
--- ----------- -------- -------- ---------- -------- -------- --------
 _  CICSKY41    OMIICICS AOC9     ACTIVE     IPSPOC0  SAOM     AOFSES01
 _  DB2SGG4     OMIIDB2  AOC9     INACTIVE   IPSPD2C  SAOM     AOFSES02
 _  DB2SG14     OMIIDB2  AOC9     MAINT      IPSPD2C  SAOM     AOFSES03
 _  IMS742CR    OMIIIMS  AOC9     INACTIVE   IPSPOI0  SAOM     AOFSES01
 D  OMSY4MVS    OMIIMVS  AOC9     AUTHFAIL   IPSPM2RC SAOM     AOFSES02
```

This produces output where the additional information includes:

- The **Description** field shows descriptive information for this session as specified in the automation policy
- The **Logical Unit** field shows the name of the source LU that was assigned by NetView upon establishing a Terminal Access Facility (TAF) fullscreen session.
- The **Password** field shows the password in the form of a string of asterisks or 'SAFPW'. This is the password used to logon to the OMEGAMON application.
- The **Timeout** field shows the maximum time to wait for a response from OMEGAMON before the request is terminated.
- The **Logon data** field shows the data that is sent to the OMEGAMON session during logon.
- The **Users** field shows a list of operators or *AUTO that have interacted with the session since it became ACTIVE.
- The **Statistics** section shows:
  - The **Total # Commands** field shows the number of commands that have been issued on this session since the session became ACTIVE. The counter is reset each time the session becomes ACTIVE.
  - The **Total # exception analysis** field shows the number of exception trap analysis commands that have been issued since the session became ACTIVE. The counter is reset each time the session becomes ACTIVE.
  - The **Total # exceptions tripped** field shows the number of exceptions that actually tripped as reported by the session since it became ACTIVE. The counter is reset each time the session becomes ACTIVE.

Monitoring Processor Status

Processor status can be monitored in detail using the processor operations 3270-type panels.

This section contains the following subsections that describe certain aspects of monitoring processor status:

- “View Status Summary of all Target Systems” on page 150
- “View One Specific Target System” on page 151

View Status Summary of all Target Systems

On a *3270 console*, issue `isqxdst` (with no parameters) on a NetView console. The Status Summary panel is displayed (Figure 38 on page 151).

```
ISQESUM SA z/OS Proc-Ops Target Status Summary      Updates: Dynamic 1
   Configuration in use : ACF

 I isqxiii C isqxcls O isqxopt  A,B view netlogs E events   Debug Mode : OFF
 S suspend /  H hold THW path       R Resume a suspended / hold THW path
 Cmd    Target System   Status 2 Focal Points - Primary: IPSFO Backup: IPUFM

        CFA             CLOSED
        CFB             CLOSED
        CFF             CLOSED
        CF1             UNKNOWN
        CF61            INITIALIZED
        CF62            INITIALIZED
        KEYA            STAGE-1 ACTIVATE COMPLETE
        KEYB            IPL COMPLETE
        KEYC            IPL COMPLETE
        LNXOLI1         NOT ACTIVE
        TEL01FE         LOAD FAILED


Enter=Static PF1=Help PF3=Exit PF4=Tgt Sys Summary PF5=Debug On/Off 3
 PF6=Roll    PF7=Up   PF8=Down PF9=Tgt HW Summary PF11=PATH Details
```

Figure 38. ProcOps Target Status Summary Panel

This panel includes the following important sections:

1
This indicates whether updates are static or dynamic (it changes automatically whenever the status changes). You can toggle to a static state with the Enter key (for example, when you want to enter a choice).

2
The status that you see for a target system represents the most severe condition that exists for that system. These values are extracted from the status, target hardware, attention, and console summary status fields of the Target System Summary panel.

3
The PF keys allow you to invoke other status panels.

View One Specific Target System

On a *3270 console*, use ISQXDST followed by the name of the target system on a NetView console. This displays the Target System Summary panel.

This panel has the following PF keys:

- The PF7 key displays the Processor Operations Interested Operator List panel (accessed with the ISQXMON command).
- The PF9 key displays the Target Hardware Summary panel. This panel provides detailed status information about the target hardware that the target system is defined on.
- The PF11 key displays the connection Path Detail panel. This panel provides detailed status information about a specific connection path.

The Target System Summary panel shows information about the target system, including:

Target System Name
The name assigned in the customization dialog to this target system.

Target System Description
Short textual description of this target system, defined in the customization dialog.

Status
(Updated dynamically) The current value of the processor operations internal variable *tstat*.

You can change this status from INITIALIZED, LOADFAILED, or IPLFAILED to IPLCOMPLETE by issuing the ISQVARS command to change the internal variable *tstat*. (For more information about specifying *tstat* as a keyword on the ISQVARS command, see the description of the ISQVARS command in *IBM Z System Automation Operator's Commands*). You would want to do this when you perform a cold start of processor operations while a target system is already running.

After the cold start, you issue the ISQXIII command to the target system that is already running so that its status becomes INITIALIZED, then you change the value of *tstat* to IPLCOMPLETE.

You may also want to change the status manually if the cause of the LOADFAILED status was corrected by using the pass-through facility of processor operations, or if it was corrected locally at the site of the target system.

You can also issue the ISQVARS command to change the internal variable *tstat* from IPLCOMPLETE to IPLFAILED. Processor operations sets a status of IPLCOMPLETE when it initiates a load of a target system and receives an operating system specific message that indicates that the operating system received a level where it is ready to work. However, neither of these operating systems provide a sufficient indication that the load process failed, preventing processor operations from changing the target system status appropriately.

You can find details of valid values for this status field in the description of the ISQXDST command in *IBM Z System Automation Operator's Commands*.

Target Hardware
(Updated dynamically) The current value of the processor operations internal variable *thwstat*.

You can find details of valid values in the description of the ISQXDST command in *IBM Z System Automation Operator's Commands*.

Attention
(Updated dynamically) The current value of the processor operations internal variable *tattn*.

You can find details of valid values in the description of the ISQXDST command in *IBM Z System Automation Operator's Commands*.

Status
(Updated dynamically) The current value of the processor operations internal variable *scstat*.

Lock Holder
(Updated dynamically) The current NetView operator that owns the lock for the target system and the connection path. Locks can be set using the ISQXLOC and ISQXUNL commands. Locks grant the exclusive use of the connection path of a target hardware and its associated target systems.

Last Significant Message
(Updated dynamically) The text of the last command response or report received from the target system. The message is not removed from the panel when the condition that generated the message is resolved.

For further details, see the description of the ISQXDST command in *IBM Z System Automation Operator's Commands*.

How to Monitor Communications Path Status

The communications paths from the processor operations focal point system to target systems are based on SNMP. You can display details of the connection for this type of path.

Press PF11 on the Target System Summary panel or Target Hardware Summary panel to display the Path Detail Panel. This displays information such as:

Support Element IP Address or Hostname
The name or IP address assigned in the customization dialog to the SE of this target hardware.

Hardware Management Console IP Address or Hostname
The name or IP address assigned in the customization dialog to the HMC of this target hardware.

Connection Authorization
The community value is shown. The authorization values are defined in the customization dialog for a selected target hardware.

Lock Holder
(Updated dynamically) The current NetView operator owning the lock for the connection path. Locks can be set and unset using the ISQXLOC and ISQXUNL command. Locks grant the exclusive use of a connection path.

SNMPv3 Security
SNMPv3 security status.

Y
The connection is SNMPv3 and authenticated by SNMPv3 User and Password.

N
The connection is SNMPv2 and authenticated by the community name (**Connection Authorization**).

SNMPv3 User
SNMPv3 User specified for the connection.

Path Status
(Updated dynamically) The status of the NetView path.

How to Monitor Alerts on a NetView Console

Alerts are high priority events that require operator action.

Issue NPDA from the command line of your NetView operator's console. The NPDA (Network Problem Determination Application) menu is displayed. You can request a dynamic display of alerts. From that display you can select an individual alert and see event detail and recommended action. For more details on the use of the NPDA panels, refer to *IBM Z NetView User's Guide*.

Monitoring with the Status Display Facility

The Status Display Facility (SDF) consists of a hierarchy of dynamically updated panels showing color-coded status conditions. It is set up by your system programmer during the customization of SA z/OS.

SDF displays the status of various resources in a z/OS system, and in other z/OS systems that specify this system as an automation focal point.

You use SDF to monitor the status of application software, WTORs, and gateways on your systems from a NetView or operator console.

The status conditions that can be displayed by SDF include those for:

- Applications and subsystems
- Monitor resources
- WTORs
- Gateways: A *gateway* is a combination of a NetView-NetView task session and two automation operators (one on each of two systems) that allows communication of messages, commands, and responses between the two systems.
- Application groups
- Spool usage problems from z/OS subcomponents
- Exceptional messages
- Processors

Because SDF can be customized, your panels may not look like those shown in this topic.

Setting Up SDF

All the resources that you need to monitor are set up by a hierarchical arrangement during customization of SA z/OS, when the system programmer:

1. Defines the SDF hierarchy.

   The SDF hierarchy indicates which automated resources are affected by status changes in other automated resources (see "How SDF Works" on page 155). The SDF hierarchy can be different from that specified in the automation control file for system startup and shutdown. Use whatever makes sense for your system.

   If SDF displays more than one system, programmers create an SDF hierarchy for each system.
2. Creates the panels that show the systems and subsystems. These panels can also be customized to display gateway sessions and WTORs.
3. Determines the status colors and other screen defaults for the enterprise or takes the defaults for these attributes.

SDF Focal Point Monitoring

Initially you need to construct an SDF screen that contains all the systems within your sysplex. This gives you an *at-a-glance* panel that is dynamically updated with the overall status of each system in your sysplex.

By selecting an individual system, you can view a more detailed display of the state of the resources on that system.

On the focal point, you can issue a command that specifies a target system. The command interacts with the target system (via XCF or RMTCMD) and gives you a full screen display as if you were logged on to the target system.

This means that from SDF you can issue DISPINFO on a target subsystem by simply pressing a key. This allows you to see detailed information about a subsystem that has a problem.

Other commands that support this kind of usage include:

- DISPERRS
- DISPFLGS
- DISPSTAT
- DISPTREE
- INGGROUP
- INGINFO
- INGREQ
- INGTHRES
- INGVOTE
- SETSTATE

All of these commands can be used from the focal point to interrogate the target system without having to log on to it.

Note: If the target system is not in the same SAplex as your focal point system, RMTCMD is used for intersystem communication. This means that there is a dependency upon VTAM being active on both the focal point and the target system for communication to be possible. If the target system is in the same SAplex, communication occurs via XCF and you do not need to have VTAM active on the target system to talk to it.

Interpreting Colors in SDF and DISPSTAT

In SDF and the DISPSTAT command dialog, subsystems appear in different colors that indicate their status. In SDF, the condition of WTORs, application groups (APGs), monitor resources (MTRs), and gateways are also indicated by color.

Refer to "Priority and Color Default Assignments" in *IBM Z System Automation Programmer's Reference* for the list of default priority and color assignments to status values. You can modify these values or define your own values when you define your own SA z/OS policy. For more details refer to "Status Display Entry Type" in *IBM Z System Automation Defining Automation Policy*.

Note: All SDF panels can be modified according to your needs.

How SDF Works

SDF uses the hierarchy established during SDF customization to propagate colors up to the highest level panel.

Figure 39 on page 155 shows a graphic representation of an example SDF hierarchy for KEY1.

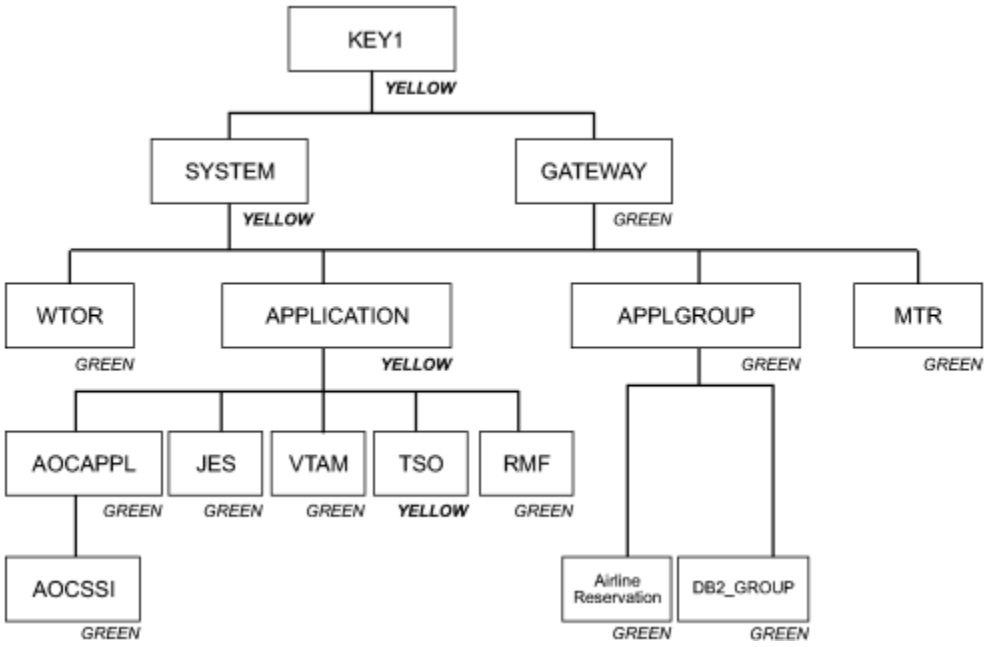


Figure 39. Example SDF Hierarchy for KEY1

Imagine that TSO appears in yellow on your screen because it is in AUTOTERM status. This color is reflected up the hierarchy to APPLICATION, SYSTEM, and KEY1, because AUTOTERM has a higher priority than the status UP (indicated by the color green). If various resources in a system have different status conditions, the one with the highest priority status controls the color of that system. This allows you to resolve issues concerning the highest priority status first, then deal with the lower priority status conditions.

Accessing SDF

You can access SDF from either a NetView session or the SA z/OS command dialogs (see “Using Command Dialogs” on page 113). Enter one of the following:

- `sdf` at a NetView command line
- `sdf` or 3 (SDF) at the command line of the SA z/OS System Operations Main Menu

This displays the highest level (or root) panel of SDF. Figure 40 on page 156 shows an example of the highest level panel for SA z/OS-supported systems. On this panel, >KEY1 IPSFM is highlighted to

indicate that it appears in reverse red on your screen, which means that a resource on KEY1 is BROKEN. All other systems may appear in green, which means UP.

Using SDF

You use SDF to monitor the status of subsystems, WTORs, APGs, MTRs, and gateways on your systems and other systems.

To monitor status, you observe the highest level SDF panel (also known as the root panel or system summary panel). Figure 40 on page 156 shows a panel with the systems of a sysplex.

```
                          SA Z/OS TEST-SYSTEMS


   KEYAPLEX        KEY1PLEX         SATPLEX         TSAPLEX         AOCPLEX

 >KEYA  IPXFG    >KEY1  IPSFM    >SAT1  IPZFA    >TSA1  IPZFL    >AOCA  IPUFA
 >KEYB  IPXFH    >KEY2  IPSFN    >SAT2  IPZFB    >TSA2  IPZFM    >AOCB  IPUFB
 >KEYC  IPXFI    >KEY3  IPSFO    >SAT3  IPZFC    >TSA3  IPZFN    >AOCC  IPUFC
                 >KEY4  IPSFP    >SAT4  IPZFD    >TSA4  IPZFO    >AOCD  IPUFD


                     S T A N D A L O N E   S Y S T E M S

                 >AOC1  IPUFG    >AOC4  IPUFJ    >AOC7  IPUFM
                 >AOC2  IPUFH    >AOC5  IPUFK    >AOC8  IPUF8
                 >AOC3  IPUFI    >AOC6  IPUFL    >AOC9  IPUF9

                               HARDWARE

                               >PROCESSORS
                                                              08/18/14 14:42
===> _____________________________________________________________________

1=HELP 2=DETAIL 3=RETURN    6=ROLL   8=ZOOM/NEXT      12=SHOW SAM/XDR SYSTEMS
```

Figure 40. Example SDF System Summary Panel

On this panel, each system is displayed in a color that reflects the highest priority status of the resources in that system. If a resource in a system changes status, the system changes color to reflect the new status. By observing this panel you can see status changes in any of your systems.

You can use the PF8 and PF7 keys to scroll between any further panels. You can tell whether this is possible because the list of PF keys at the bottom of the panel shows 8=ZOOM/NEXT. Do not position the cursor on a system because this invokes the zoom function of the PF8 key.

You select a system to investigate by moving the cursor to that system using either the Tab key or the mouse.

Imagine that `>KEY1 IPSFM` is displayed in reverse red. This indicates that a resource on system KEY1 is in the BROKEN status. To view more information about the broken resource on this system, select KEY1 by moving the cursor to it and pressing PF2 to display a Detail Status Display panel. This gives a detailed description of the broken resource with the highest status priority. This information is also shown for all the resources that are defined in that domain on successive panels that are ordered from the highest status priority to lowest . See “How to Check Resource Status Conditions” on page 157 for more information about the Detail Status Display 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. To view further details, select an item and press either PF2 or PF8:

PF2
This displays the Detail Status Display panel for the component with the highest status priority.

PF8

This displays a status panel that lists all the components in order of their status priority and in the colors that represent their individual statuses.

Select a component in this summary list and press PF2 to display its Detail Status Display.

There may be a list of PF keys that invoke SA z/OS command dialogs for these components at the bottom of the panel, such as:

```
13=EXPLAIN  17=SETSTATE 18=INGVOTE 19=INGREQ             23=INGLIST 24=INGINFO
```

Press the Shift and PF key to access these command dialogs, for example, Shift+PF5 for PF17, SETSTATE.

Note: After returning from executing a command (for example, PF2 to display a Detail Status Display panel), the cursor position is restored to the position prior to the command execution.

How to Check Resource Status Conditions

Normally when you use SDF to monitor resources, you observe the highest level panel until you see a system change color.

This color change indicates a change in status in one of the resources in the system. To determine which resource, and to see more information on the status change:

1. Select the system (or subsystem) that shows a status change by moving the cursor to it using the Tab key or the mouse.
2. Press PF2. This takes you to the Detail Status Display panel for the resource that caused the system to change color.

   If more than one resource changes status at the same time, SDF shows the information for the system with the highest priority status first. You can press PF8 to page through the Detail Status Display panels for other resources on the system.

In the example from Figure 40 on page 156, you select KEY1 (the system in reverse red). When you press PF2, you see the Detail Status Display panel for HUGC2APL0 (Figure 41 on page 157), which is the resource with the highest priority because it is BROKEN.

```
        1 of 242         ---- Detail Status Display ----         03/04/09 12:34:39

        Component . . . : HUGC2APL0                  System . . . . : KEY1

        Color . . . . . : RED                        Priority . . . : 55

        Date  . . . . . : 03/04/14                   Time . . . . . : 10:20:35

        Reporter  . . . : GATIPSFM                   Node . . . . . : IPSFM

        Jobname . . . . : HUGC2APL0

        Reference value : C3D57D761DF1520E

        User data . . . :

        AOF571I 10:20:05 : HUGC2APL0 SUBSYSTEM STATUS FOR JOB HUGC2APL0 IS BROKEN -
        AT SYS-OPS RECYCLE , JOB AND ALL DEPENDANTS WILL NOT BE STARTED


===>
1=Help 3=Return 4=Delete 6=Roll 7=Up 8=Down 11=Bottom 12=Top
```

Figure 41. Example SDF Detail Status Display Panel for HUGC2APL0

This panel also shows the automation or human operator that sent the notification and gives the priority number that corresponds to the automation status. Priority numbers determine which color the resource

is displayed in and the order of the Detail Status Display panels (the lower the number the higher the priority).

In this example, the panel is the first of 242 Detail Status Display panels as indicated at the beginning of the panel header (1 of 242). Each panel corresponds to a resource on system KEY1 for which SDF maintains detailed information. The panel retains its current index value (such as 1 in this case), which represents its position in the sequence of panels. This index remains unchanged unless it is the final one in the list for example 242 and the total number of panels is reduced. Although the index remains fixed, the content of the panel might update dynamically to reflect the most recent status details for the resource currently ranked at that index. For example, with index 1, the panel always display the latest information for the highest-priority resource.

You can use the following PF keys:

PF3
Returns to the previous panel.

PF4
Deletes the panel you are currently on. (The record is physically deleted.)

Note: You should not normally use this PF key.

PF7
Scrolls back up the Detail Status Display panels in order of descending status priority.

PF8
Scrolls down the Detail Status Display panels in order of descending status priority.

PF11
Moves to the Detail Status Display panel for the resource with the lowest priority, that is, the last panel in the Detail Status Display stack.

PF12
Moves to the Detail Status Display panel for the resource with the highest priority, the first panel in the Detail Status Display stack.

Note: You can customize the PF keys. For more details, see "Status Display Facility Definitions" in *IBM Z System Automation Programmer's Reference*.

Deleting SDF Records

The SDFCONF command is assigned to the PF4 key on the Detail Status Display panel to delete an SDF record.

This is useful because it prompts you for confirmation before performing the actual deletion and leads to a Delete Confirmation panel, as shown in Figure 42 on page 159. If you do not want the prompt panel to appear, then add ",VERIFY=NO" to the end of the SDFCONF command.

```
AOFKNSSC                  SA z/OS  -  SDF Status Display
Domain Id . : IPZNA     ------- Delete Confirmation ------    Date . . : 06/12/14
Operator Id : OPER1               System = SAT1              Time . . : 13:13:50


Press ENTER to delete this SDF detail entry.

Component. . . . : IMSD1BMP(SUBSYS)
Reference value. : IMSD1BMP
Date / Time. . . : 06/11/14 13:30:37
Reporter . . . . : AUTWRT13
Node . . . . . . : IPZNA

AOF571I 13:30:36 : IMSD1BMP SUBSYSTEM STATUS FOR JOB IMSD1BMP IS BROKE
N - ABENDED, CRITICAL THRESHOLD EXCEEDED







Command ===>
   PF1=Help              PF3=Return          PF5=SDFDEL         PF6=Roll
                                                               PF12=Retrieve
```

Figure 42. Delete Confirmation Panel for an SDF Record

On this panel you can:

- Press Enter to confirm the deletion of the SDF detail entry
- Press PF3 to cancel the deletion request and to return to the previous panel
- Press PF5 to delete the entry from SDF only using the internal SDFDEL command

Note: Deleting an exceptional message from SDF removes the message from all other interfaces where it is shown.

Config Refresh Monitoring

The tree and panel definitions are defined in the members INGTCFG:

```
   1 INGCFG
      2 &SDFCsaplex.
         3 AGENT
```

and INGPCFG:

```
P(INGCFG,*,*,SYSTEM,SYSTEM,,*,*
:
BODY(INGCFG,05,*-6,6,03,79,,VA)
BH(01,L,S,T,N,68,79)
CELL(02,09,N,V)
CELL(11,26,N,B)
CELL(28,35,N,C)
:
```

Both members are included in the members AOFTREE and AOFPNLS. The subsequent snapshots base on the following CNMSTYLE sheet definitions:

```
COMMON.AOF_AAO_SDFCSAPLEX.0 = 2
COMMON.AOF_AAO_SDFCSAPLEX.1 = KEYAB_PLEX KEYC_PLEX
COMMON.AOF_AAO_SDFCSAPLEX.2 = AOC1_PLEX
```

The definition of the compound variables follows the same rules as described in "Step 21: Configure the Status Display Facility (SDF)" of *IBM Z System Automation Planning and Installation*. The sample above defines three SAplexes

```
SAplex           Sysplex   System(s)
---------------  --------  ---------------
AOC1_PLEX        AOC1PLEX  AOC1
```

```
KEYAB_PLEX          KEYAPLEX   KEYA,KEYB
KEYC_PLEX           KEYAPLEX   KEYC
```

on two physical sysplexes. You may group the SAplex names like in the sample by the sysplex membership or by other criteria. Since only active systems are displayed on the INGPCFG panel, you can share the definition across all systems no matter whether the information is available or not.

The BODY statement of panel INGPCFG defines two logical columns. To keep the column header of each logical column identical, we cannot use the BH statement for displaying the header. Otherwise, we would see the number of systems on the header of the right column. Instead, we use TF and TT statements for defining each header. The associated BH statement is now located in the upper-right corner of the screen and just displays the number of systems that are active.

Figure 43. INGPTOP Panel (KEYC is not active)

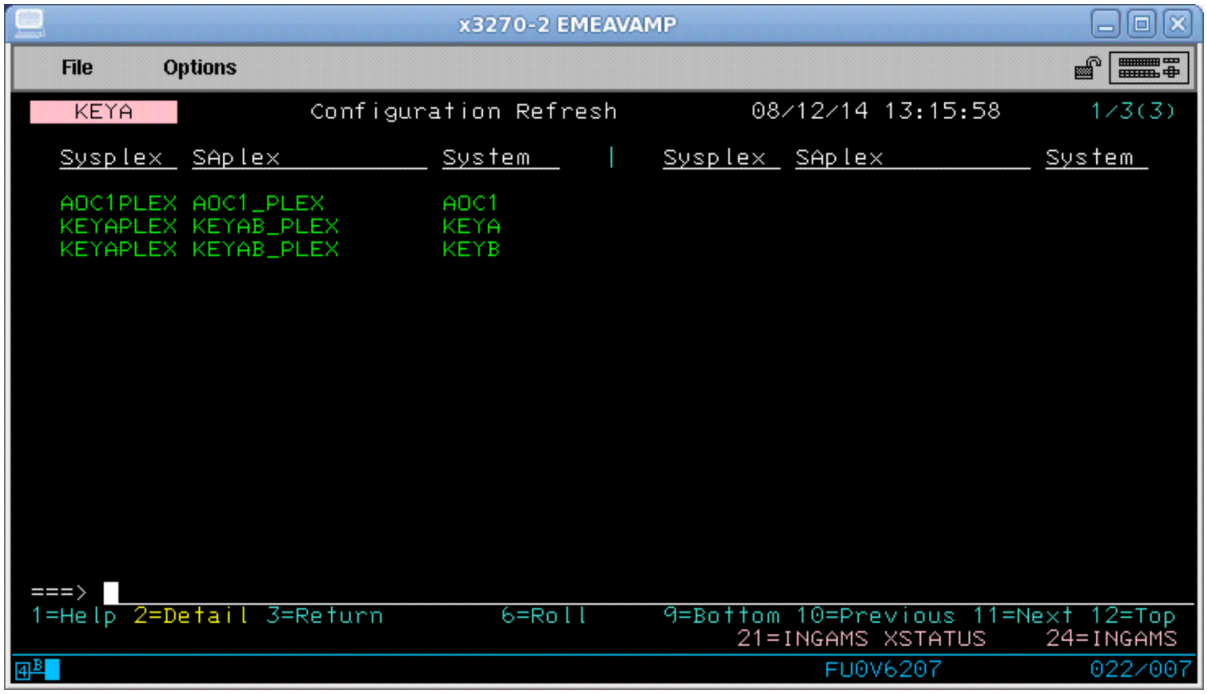


Figure 44. INGPCFG Panel (KEYC is not active)

Refer to "AOFTREE" in *IBM Z System Automation Programmer's Reference* for the general usage of the AOF_AAO_SDFCxxx.n variables.

Working with the Looping Address Space Monitor

The Looping Address Space Suppression procedure is an active monitor that runs inside an APL. The suggested naming for the APL is to give it a subsystem name of LOOPSUPP.

Stopping and Starting the Procedure

The procedure should normally be running 24 hours a day, but if you need to stop it you can use INGREQ and INGVOTE to ask for it to be shutdown or restarted like a normal APL or both.

Changing its Automation Flag

The execution of the Monitor within the APL is controlled by a minor resource called MONITOR. The default for it is set to LOG. You can use INGAUTO to dynamically change its setting.

Specify LOOPSUPP as the subsystem and MONITOR as the minor resource. Setting NO disables the monitor, setting LOG restricts it to only logging its active recovery commands and setting YES lets it run fully.

Changing the Automation Flags for specific address spaces

Before it takes any recovery action for an address space, the code checks an automation flag specific to the address space. This is a minor resource under the LOOPSUPP system and is named `category.jobname`.

The category is the recovery category derived from the categorization process. It is printed out in the monitoring report in the Netlog if you need to look it up.

Setting NO turns recovery off, setting LOG only permits non-disruptive recovery and setting YES permits full recovery.

You can also set a flag for just the category, which then affects all address spaces that fall into that category that do not have specific automation flag settings. This enables you to, for example, turn all TSO_USER recovery off.

If you need to make a quick change, you need to use INGAUTO to change the automation flag.

Warning Messages

When the loop suppression monitor issues a warning message (ING601E) in response to a WARN directive, the resulting message is issued to the Netlog as classes 40 and 44. Any notify operator with either of these classes active receives it. If the SYSOPER notify id is subscribed to either class, it is issued to system consoles and the SYSLOG as well.

Warning messages are also posted to SDF, under the MVSESA qualifier. They may be viewed through SDF. The Monitor automatically clears the message if it is shut down or it is no longer detected as being looping. The color of the message (Yellow, Pink, Red) is determined by how many consecutive passes it has detected on (<10, <20, 20+).

Viewing the Monitoring Report

The report is visible in the Netlog with message ID ING600I.

Viewing the Diagnostic Reports

The reports appear in the Netlog with the message ID ING602I a few seconds after the monitor report that they are associated with.

A typical diagnostic report looks like the following:

```
<xmp>
ING602I DIAGNOSTICS FOR REXXLOOP (ASID 00D8)
Messages from the Inspect tool:
KM3IN008I GRANULARITY SET TO 0X00000B40
Requested 1000 samples at 5us intervals
Took 1000 samples of which 1000 were valid.
Program: EXEC
  TCB  :          @ 008E3390 (100.0 JOB)
  Lmod : IRXINIT  @ 0C467000 (100.0 JOB 100.0 TCB)
  CSECT: *-UNKN-* @ 0C467000 (100.0 JOB 100.0 TCB, 100.0 LMOD)
                    Offset   Percentage
                    -------- ----------
                   +00000B40       73.3
                   +00009240        6.6
                   +00000000        5.1
                   +00009D80        3.4
                   +00006540        3.1
                   +000021C0        2.4
                   +00007080        2.2
                   +00001680        1.7
                   +0000A8C0        1.1
                   +0000B400        0.7
                   +00007BC0        0.1
</xmp>
```

The first line gives us the jobname and the ASID.

Then follow one or more messages that were produced by the OMEGAMON inspect tool. These all have message ids like KM3INnnnn.

The next two lines give details of the number of inspection samples that were requested, their interval, the number that were taken and the number that were valid. The sampling values used are the default for the inspect tool.

The inspect tool works by observing the location of the program counter within the target address space. This tells us which opcode within the program it is about to execute. By repeatedly sampling this, OMEGAMON can build up a profile of the program short term execution flow. If all the samples are within

a few bytes of each other then it is in a very tight loop. If they are more spread out, then the body of the loop encompasses a large part of the program's control structure.

Note that each TCB has its own program counter, so for a multi-threaded program you get multiple CPU traces.

If the program has made a cross memory call, the load module details are proceeded by a line indicating the address space (and its ASID) that the load module is within.

For the report above:

- The address space is executing a program called EXEC. This is the normal way of calling the REXX intepreter.
- There is a single Task Control Block (TCB), which is at address 008E3390. 100% of the samples indicated that the TCB was being dispatched.
- The Load Module is IRXINIT, also the REXX Interpreter, and it is loaded at address 0C467000. 100 percent of the samples found active execution within it. If execution is passing between two load modules, the load module, CSECT and OFFSET records are repeated for each load module.
- The name of the CSECT cannot be determined but its load address is the same as the load modules, so we are in the mainline of the load module. 100 percent of the samples placed the program counter within the CSECT.
- The OFFSET statements need to be read in conjunction with the KM3IN008I message from the Inspection tool. The sampling found the program counter in a number of different offsets. For reporting purposes they have been grouped together into blocks. Each block is of the width stated in the granularity message (B40 bytes in this case) and is identified by its base offset from the start of the CSECT (so +0000000, +0000B40, +00001680 and so on). The list of blocks is sorted by percentage of the samples that were found within that block. This information can be used to work out what the program is doing.

Monitoring and Controlling a Sysplex

SA z/OS provides a command interface to monitor and operate sysplexes. It helps operators to manage critical sysplex resources like coupling facilities and CF structures, couple data sets, and more. You can benefit from reduced sysplex operations complexity leading to higher sysplex availability and greater productivity.

The following section describes the panels of the system operations commands that support you in working with sysplex specific automation tasks. For details about the syntax and parameters of these commands, refer to *IBM Z System Automation Customizing and Programming*.

How to Work with INGPLEX

SA z/OS provides the INGPLEX command to monitor and control the couple data sets and coupling facilities of your sysplex.

If you enter INGPLEX in the command line without any parameters, the INGPLEX Selection Panel is displayed. To launch other command dialogs that show sysplex-related information, specify the number or the function and press Enter.

The subcommands support the following functions:

1 SYStem
Displays the target sysplex name, its GRS mode and its systems. For each system, details about the system name and status are shown.

2 CONsole
Displays details about the target sysplex. For example, the master console name, WTO and WTOR buffer utilization, and the number of queued messages.

3 CF
Displays the coupling facilities in the sysplex. For each coupling facility the structures and paths can be displayed, and the coupling facility can be drained and enabled.

4 CDS
Displays the couple data sets in the sysplex. For each couple data set you can obtain detailed information, and you can switch the couple data sets or allocate new alternate couple data sets.

6 IPL
Collects IPL information.

7 DUMP
Displays the DUMP submenu.

10 INGSTR
Displays all the structures in the INGSTR panel. See "How to Work with INGSTR" on page 179

These functions are described in more detail in the following sections.

INGPLEX SYStem

INGPLEX SYStem displays the target sysplex name, its GRS mode and its member systems.

- The **Sysplex** field shows the name of the sysplex.
- The **GRS Mode** field shows the GRS mode of the target system. The mode can be either STAR or RING.
- The **CMD** column allows you to specify one of the command codes to display more information. Type the appropriate letter next to the resource name, and press Enter.
- The **System** column shows the name of the system.
- The **Status** column shows the status of the system.
- The **Monitor Timestamp** column shows the last time stamp recorded for status monitoring on this system.
- The **INTERVAL** column shows the system failure detection interval in seconds. This interval is the time XCF lets elapse without a status update before assuming that the system failed.

The last three columns contain configuration data of the SFM policy (if applicable).

- The SSUM Action field shows the SSUM action. It can be one of the following:
  - ISOLATE
  - DEACTIVATE
  - RESET
  - PROMPT
  - N/A
- The SSUM TIME field shows the SSUM interval as specified in the current SFM policy.
- The SSUM WEIGHT field shows the SSUM weight specified in the current SFM policy. This weight is used in sysplex configurations after a signalling connectivity failure.

Use the PF11 key to view WLM-related information to the right:

- The **Timestamp** field shows the last time stamp when capacity data was queried from WLM.
- The **SUs Total** field shows the number of free SUs in the last 10 minutes.
- The **SUs Used** field shows the number of used SUs in the last 10 minutes.
- The **Resource** field shows the number of SU-consuming resources with a desired state of AVAILABLE.
- The **SUs Exp.** field shows the number of free SUs, taking into account the resources that SA z/OS is about to start or stop.

To display more detailed information for one of the systems, enter the following command codes in the **Cmd** field next to it:

C
Displays the online or offline status of one or more processors and any vector facilities, or ICRFs attached to those processors

E
Displays the timer synchronization mode and ETR ports

I
Displays IPL information

O
Displays IOS-related configuration information

S
Displays the number of megabytes of central and expanded storage assigned and available to the system

D
Displays the device number of one or more inbound or outbound signalling paths that XCF can use and information about inbound or outbound XCF signalling paths to this system

T
Displays detailed signalling path information for all coupling facility structures

INGPLEX CONsole

INGPLEX CONsole displays the following information for the target sysplex:

- The name of the master console
- WTO & WTOR buffer utilization
- Number of queued messages (replies) of various types
- Awaiting mounts
- Operator requests and list of consoles (name, status, authority, number of WTOR buffers, UD, device, system, ALTGRP, MSCOPE)

Use one of the following command codes to get more information for the selected console or consoles.

D
Displays details for the console

R
Displays current requests for the console

The fields on the command dialog panel display the following information:

- The **Sysplex** field shows the name of the sysplex.
- The **Message Buffer Usage** field shows the limit of the number of WTO message buffers allowed outstanding.
- The **Awaiting Replies** field shows a decimal number of messages awaiting replies.
- The **Immediate Action** field shows a decimal number of outstanding immediate action messages (with descriptor codes 1 or 2). If the number is greater than 99999, asterisks appear in this field.
- The **Critical Action** field shows a decimal number of outstanding critical eventual action messages (with descriptor code 11). If the number is greater than 99999, asterisks appear in this field.
- The **Master Console** field shows the name of the master console.
- The **Reply Buffer Usage** field shows the limit of the number of WTOR message buffers allowed outstanding. The maximum value of *yyyy* is specified by the RMAX parameter in the CONSOL*xx* parmlib member.
- The **Eventual Action** field shows a decimal number of outstanding eventual action messages (with descriptor code 3). If the number is greater than 99999, asterisks appear in this field.
- The **Awaiting Mounts** field shows a decimal number of outstanding mount requests.
- The **Operator Requests** field shows a decimal number of outstanding requests for operator intervention.

The following details are shown for each MCS console that has been defined.

If the MSCOPE information does not fit on the primary screen, the PF10 function key is available for you to toggle between the primary panel and the panel showing only MSCOPE information.

- The **Console** column shows the name of the console as specified in the CONSOLxx parmlib member.
- The **Status** column shows the status of the console. The following values can occur:

  HARDCOPY
  Hardcopy log. This condition is indicated when the console is active on the system where the command processes.

  ACTIVE
  Active console.

  ACTIVE-P
  In the process of becoming an active console. This condition will be indicated only when the console is active on the system where the command is processing.

  MASTER
  Master console

  INACTIVE
  Inactive console

  INACT-P
  In the process of becoming a non-active console. This condition will be indicated only when the console is active on the system where the command is processing.

  PROB-DET
  The active system console is in the problem determination mode. PD is indicated only for the system console.

  SUBSYS
  Subsystem-allocatable console
- The **AUTH** column shows which commands can be entered from this console. The following values can occur:

  ALL
  Any INFO SYS, IQ or CONS command can be entered from this console.

  CONS
  INFO commands and any commands from the console command group can be entered from this console.

  INFO
  Any command from the informational command group can be entered from this console.

  IO
  INFO commands and any commands from the I/O Control command group can be entered from this console.

  MASTER
  The specified console is authorized to enter any operator command.

  NONE
  This console has no command authority.

  SYS
  INFO commands and any commands from the system control command group can be entered from this console.
- The **NBUF** column shows the number of WTO message buffers currently queued to this console. If *nnnn* is greater than 9999, asterisks (****) appear in this field.
- The **UD** column shows whether this console is receiving messages with the UD attribute.
- The **Device** column shows the name of the console as specified in the CONSOL*xx* parmlib member.
- The **System** column shows the system name of the active console.
- The **ALTGRP** column shows the alternate group defined for this console.

- The **MSCOPE** column lists the name of the system or systems that this console is receiving unsolicited messages from. Note that these systems might be different from the system where this console is physically attached.

INGPLEX CF

With INGPLEX CF you can:

- Drain or enable a selected coupling facility
- Display the paths and structures to a selected coupling facility

This panel displays all coupling facilities in a sysplex including particular storage information of each coupling facility. On this panel you can select one of the following actions by entering the command code next to the coupling facility:

D (Drain CF)
Releases a coupling facility from its structures and connections in order to remove it from the sysplex for maintenance.

E (Enable CF)
Integrates a coupling facility into a sysplex.

M (Start or stop maintenance mode)
Puts the coupling facility into or takes it out of maintenance mode.

P (Display sender paths)
Displays the paths from the systems in the sysplex having a connection defined to the specified coupling facility. For each path the physical and the logical status are displayed.

S (Display structures)
Shows all structures allocated in the specified coupling facility.

These actions are described in detail in the following sections.

Drain CF

With this function, you can remove a CF from the sysplex, for example, for maintenance purposes. You can perform the following sequence of tasks:

1. Display information for all allocated structures of the CF.
2. Put the coupling facility into maintenance mode if the MAINT function is available.
3. Rebuild all rebuildable structures on another CF and delete instances of structures on the target CF that are being duplexed on another CF.
4. Force the deletion of structures that have no active connectors and cannot be rebuilt.

   Note: There are structures that you can neither rebuild nor delete with the force action. These include the structures that have at least one active connector and do not support rebuild. To remove such structures first disconnect all active connectors, and then delete the structure manually if it is persistent, or has persistent connections.
5. When the coupling facility is not in maintenance mode, disconnect the coupling facility from the systems that it is connected to.
6. Disconnect the CF from the systems that it is connected with.

The DRAIN option ensures that these actions are performed in the correct order, as specified above.

Depending on the status of the CF and its allocated structures, you can execute one of four actions with the program function keys:

REBUILD (PF10)
Starts the rebuild of structures that can be rebuilt on another CF. Therefore, a rebuild is only initiated for structures whose preference list contains more than one CF.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems

that have z/OS 2.8 or above; it must have been enabled by formatting the CFRM couple data sets with the specification

```
ITEM NAME(SMREBLD) NUMBER(1)
```

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

The REBUILD action also deletes all structure instances on the target CF that are being duplexed on another CF.

FORCE (F5)
Forces the deallocation of structures with one of the following conditions:

- No connection exists.
- No alternate CF for structure with no active connections.
- No alternate CF for structure with no connections.

This action is only available after all structures that can be rebuilt have been rebuilt.

***MAINTON (F5)**
Puts the coupling facility into maintenance mode.

Note that this function is only available before you start the rebuild process. Once you have started the rebuild process and you want to put the coupling facility into maintenance mode you need to issue the command INGCF MAINT *cf_name* MODE=ON.

DRAIN (F4)
Disconnects the coupling facility from its connected systems by setting the sender paths OFFLINE.

This action is only enabled after all structures of the target CF have been removed to another CF or deallocated. Note that structures that have active connectors but do not support rebuild cannot be removed with PF10 or F5. They must be deallocated manually before execution of this step is enabled.

Example of Draining a CF

The status of the CF (NORMAL) and the authorization type of the operator (ALL) are displayed on the right of the INGCF DRAIN panel. The main part of the panel consists of a list of the structures allocated in CF1 and their conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

The only action that is enabled is REBUILD with PF10. Pressing PF10 leads to a confirmation panel. After PF10 has been pressed, the structures are in the process of being rebuilt.

After the rebuild has been completed, a panel is displayed that shows that one structure could not be rebuilt because no alternate CF is specified in its preference list. The REBUILD status is no longer available. Instead, the FORCE action (F5) is offered because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure. Pressing F5 displays a confirmation panel similar to that for REBUILD.

Pressing PF10 on the confirmation panel and refreshing the command dialog after the action has been completed results in a panel that shows that no more structures are allocated in the coupling facility, so the coupling facility can be released from the connections with the systems of the sysplex. Consequently, INGCF DRAIN enables the DRAIN action (F4). After completion of that action, the status of the coupling facility changes to DRAINED. Because the coupling facility is no longer connected to any system, it can be inactivated. After pressing F11 the status of the coupling facility changes to INACTIVE.

Example of Draining a CF Using MAINTMODE

In the following example, a coupling facility is drained:

1. The coupling facility is put into maintenance mode.
2. All of its structures that can be rebuilt are rebuilt on another coupling facility, and duplexing is stopped.

3. Deletion is forced for all structures that have no active connector and cannot be rebuilt.
4. The coupling facility is made inactivate.

   Note: It is no longer necessary to turn the sender paths offline when the coupling facility is in maintenance mode, regardless of whether or not all systems in the sysplex run z/OS 1.9 or later.

When you issue INGCF with the DRAIN option, you can specify the coupling facility to be drained, for example, by entering INGCF DRAIN CF02. If you do not specify a coupling facility name, INGCF displays a selection panel with all coupling facilities that are defined in the sysplex.

The status of the coupling facility (NORMAL) and the authorization type of the operator (ALL) are displayed on the right side of the panel header. The main part of the panel consists of a list containing the structures allocated in CF02 and their conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

Two actions are enabled, MAINTON with F5 and REBUILD with F10. Pressing F5 calls the confirmation panel for starting the maintenance mode.

After pressing F10 on the confirmation panel, the main panel shows the new status (NORMAL MAINTMODE) and leaves only F10 for the rebuild process. Pressing F10 calls the confirmation panel for the rebuild process.

After pressing F10 on the confirmation panel and the rebuild is complete, you can refresh the command dialog with F9.

One structure could not be rebuilt because no alternate coupling facility is specified in its preference list. The REBUILD action is no longer available. Instead, the FORCE action (F5) is available because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure.

Pressing F5 calls a confirmation panel similar to that for REBUILD. Pressing F10 on the confirmation panel and refreshing the command dialog after the action has been completed results in an empty panel and the status of the coupling facility has changed to MAINTMODE.

Because no more structures are allocated in the coupling facility, it can be inactivated. After pressing F11 the status of the coupling facility changes to INACTIVE.

Enable CF

The ENABLE function of the INGCF command is intended to support the integration *and* reintegration of a coupling facility into a sysplex. With this option, you can:

1. Activate the target coupling facility.
2. Connect the systems of the sysplex with the coupling facility.
3. Take the coupling facility out of maintenance mode when it is in this mode and the MAINT function is supported.
4. Switch to another CFRM policy if the target CF is not defined in the active policy and a suitable policy is available.

   A suitable CFRM policy must contain:

   - A definition of the target CF
   - Appropriate definitions for every active CF and every allocated structure
5. Rebuild all structures on the target CF whose preference list starts with this CF, unless this is excluded by other requirements.

The ENABLE CF option ensures that these actions are performed in the correct order, as specified above.

The following function keys support these actions:

***ACTIVATE (F11)**
This action activates the CFCC (Coupling Facility Control Code) through the BCP internal interface by an ACTIVATE command.

Note: This function key is unavailable when running on a z/OS image that runs under z/VM®.

***ENABLE (F4)**
Sets the sender paths of all systems of the sysplex to ONLINE. This action is enabled when the CF is active.

***MAINTOFF (F5)**
Takes the coupling facility out of maintenance mode.

***SWITCH (F5)**
Switches to another CFRM policy when the target CF is not defined in the active CFRM policy and a suitable policy is available. When there is more than one suitable policy you can choose one of these from a selection panel.

A CFRM policy is suitable when it contains:

- A definition of the target CF
- Definitions for every active CF and every allocated structure

This action is only available when the target CF is active, but not defined in the current CFRM policy.

***POPULATE (PF10)**
Starts a rebuild process where all structures that have the target CF at the beginning of their preference list, but are currently allocated on another CF, are allocated on the target CF.

This action requires that the CF is enabled, connected to all members of the sysplex, and defined in the current CFRM policy. The action is offered whenever INGCF ENABLE detects that a structure is not allocated on the target CF although it is the preferred CF of that structure.

Note: When you have drained a coupling facility with INGCF DRAIN and then reintegrate it with INGCF ENABLE, be aware that the set of structures that are allocated on the target coupling facility after population can be different from the original set before the draining. Typically, this happens when the original set does not contain exactly those structures that have the target coupling facility at the first position in their preference list.

Note that these actions can only be performed when INGCF ENABLE is called in full mode. In line mode, only the display function is available.

In the following example, a coupling facility that has already been activated is reintegrated into the sysplex in two steps:

1. The coupling facility is connected to all systems of the sysplex.
2. All structures that have the target coupling facility as the first coupling facility in their preference list are allocated on the target coupling facility.

If you issue INGCF with the option ENABLE, you can specify the coupling facility to be reintegrated, for example by entering `INGCF ENABLE CF02`. If you do not specify a coupling facility name, INGCF shows a selection panel with all coupling facilities that are defined in the sysplex.

After selection of CF02, INGCF displays a panel that shows that the selected CF has already been activated manually, therefore its status, as shown on the right of the panel, is `DRAINED`. The authorization type of the operator (`ALL`) is also displayed on the right of the panel. The main part of the panel is empty because no structures are allocated in CF02. The only action that is activated is ENABLE with PF4. If you press PF4 a confirmation panel is displayed.

After pressing PF10 on the confirmation panel, the command dialog shows that the status has changed to `NORMAL`, and you can populate the CF with PF10. This implies that the target CF is defined in the active CFRM policy.

The structure list contains three entries with the condition '`Structure is currently allocated in CF01.`'. These are the structures that are currently allocated in CF01, but have CF02 at the first position in their preference list.

Pressing PF10 populates the CF, and the refreshed panel shows that the POPULATE action is no longer available because all structures whose preference list starts with CF02 are now allocated in CF02.

Display Sender Paths

With this function, the paths from the connected systems to the specified coupling facility are displayed.

The last sender path of each system can only be set to OFFLINE when no more structures are allocated.

The following command codes are available:

F
Sets the sender path OFFLINE.

N
Sets the sender path ONLINE.

The fields on the command dialog panel display the following information:

- If you have issued INGCF with the PATH parameter, the **Coupling Facility** field is an input field. To display the path list of another coupling facility specify the name of the coupling facility in this field and press Enter.
- The **Allocated Structures** field shows the number of allocated structures.
- The **Permission** field shows your authorization level.
- The **System** column contains the names of the systems that are connected to the target CF.
- The **CHPID** column shows the IDs of the sender channel paths.
- The **Physical** column shows the status of the sender channel paths.
- The **Logical** column shows the logical status of the paths to that CF.
- The **Type** column shows the type of the sender channel paths.

Display Structures

The STRUCTURE function of the INGCF displays the allocated structures of a coupling facility. You can initiate a rebuild or deallocation of a selected structure if the conditions for these actions are satisfied.

- If you have specified INGCF with the STR parameter, the **Coupling Facility** field is an input field. To display the structure list of another coupling facility, specify the name of the coupling facility in this field and press Enter.
- The **Include Condition** field is an input field. By entering Yes or No in this field you determine whether the conditions of the structures are displayed in the **Structure** column.
- The **Permission** field shows your authorization level. There are two possible values, ALL and DISPLAY. DISPLAY indicates that you can only use the display functions. ALL indicates that you can also rebuild and delete structures.

  You can specify an action code before every structure entry. The codes you can enter depend on your authorization level
- The **Structure** column shows the names of the structures.
- The letter P in the **P** column indicates that policy changes are pending for the structure.

  A structure has policy changes pending when it was allocated at the time of a CFRM policy switch, and XES could not bring the switch into effect for that structure. One reason for a pending policy change is that the old and the new policy define the structure differently, for example, with different preference lists.
- The **D** column indicates the type of duplexing that is possible. The following values are possible:

  U
  User-managed duplexing

  S
  System-managed duplexing

  B
  User-managed and system-managed duplexing

- The **Condition** column shows the status of the structures. You can switch the display of the conditions on and off with the **Include Condition** field.

The following command codes are available:

D
Displays detail information about the structure.

***F**
Forces the deallocation of the structure if it has one of the following conditions:

- `No connection exists.`
- `No alternate CF for structure with no active connections.`
- `No alternate CF for structure with no connections.`

When you try to force the deallocation of a structure that can be rebuilt, an error message is issued.

***P**
Stops duplexing of the selected structure.

***R**
Starts the rebuild of the selected structure. Depending on the PENDING status, the automation starts the rebuild with a different LOCATION parameter (PENDING uses the parameter LOCATION=NORMAL, otherwise LOCATION=OTHER). A rebuild with the parameter LOCATION=OTHER is only initiated for structures whose preference list contains more than one coupling facility.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems that have z/OS 2.8 or above. It must have been enabled by formatting the CFRM couple data sets with the specification

```
ITEM NAME(SMREBLD) NUMBER(1)
```

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

INGCF STRUCTURE accepts a rebuild request for structures with the condition '`No active connection exists.`', but *deallocates* them. See the section "Structure Conditions" in the description of the INGCF command in *IBM Z System Automation Operator's Commands*.

The rebuild function works differently depending on the status of the structure (PENDING calls LOCATION=NORMAL, otherwise LOCATION=OTHER).

***S**
Starts duplexing of the selected structure.

There are two methods for duplexing, user-managed and system-managed duplexing. User-managed duplexing is supported for all release levels. System-managed duplexing is only available when all systems in the Parallel Sysplex have been upgraded to z/OS 1.2 or later with APAR OW41617, and appropriate APARs listed in the CFDUPLEX PSP bucket (for more information, see *System-Managed CF Structure Duplexing*, GM13-0103-03). System-managed duplexing must have been enabled by formatting the CFRM couple data sets with the specification

```
ITEM NAME(SMDUPLEX) NUMBER(1)
```

System-managed duplexing is only performed when the requirements for user-managed duplexing are not met. This applies, for example, to structures without active connectors.

Starting the duplex rebuild of a structure requires at least the policy entry allowing the duplex rebuild of the structure. If there is no entry the duplex rebuild is disabled. The other requirements depend on the type of the duplex rebuild. When all connectors to a structure allow user-managed duplex rebuild, this type takes precedence over system-managed duplex rebuild. However, user-managed rebuild also requires at least one active connector. Thus, when the operator starts the duplex rebuild for a structure allowing user-managed duplex rebuild as well as system-managed rebuild but without

having active connectors, XCF tries to initiate a system-managed duplex rebuild. System-managed duplex rebuild has the following requirements:

- System-managed rebuild must be supported by all connectors.
- The structure must be allocated in a coupling facility supporting system-managed duplexing and another coupling facility supporting system-managed duplexing must be defined in its preference list.
- The CFRM couple data set must support system-managed duplex rebuild and the structure must not have a policy change pending.
- The structure must be defined in the active CFRM policy when any connection state is not active.

INGPLEX CDS

The CDS function displays information about all the couple data sets in the system, including details of the corresponding policies. For every CDS type that is required by the implementation INGPLEX CDS allows the operator to:

- Switch from the primary to the alternate CDS
- Define a new alternate CDS
- Change the active policy (if applicable)

This panel header contains sysplex-related information about the system that the INGPLEX command was executed on. The details are as follows:

- The **System** field shows the name of the system.
- The **Interval** field shows the system failure detection interval in seconds. This interval is the amount of time XCF lets elapse without a status update before assuming that the system failed.
- The **OPNotify** field shows the number of seconds that XCF waits before notifying the operator of a potential system problem.
- The **Maxmsg** field shows the default value for the maximum amount of kilobytes of message buffer space. This default value is used when MAXMSG is not specified on SETXCF START commands.
- The **Cleanup** field shows the number of seconds that XCF waits for cleanup of members.
- The **Retry** field shows the default value for the retry limit. This value is used when the RETRY keyword is not specified on SETXCF START commands.
- The **Classlen** field shows the default length (in bytes) of messages allowed for a transport class. This value is used when CLASSLEN is not specified on the SETXCF START CLASSDEF command.
- The **Max CFlevel** field shows the maximum CFLEVEL supported by this system. This system can connect to a coupling facility with a higher CFLEVEL than the value of **Max CFlevel** but would not be enabled to use any functions supported by the higher level coupling facility.
- The **COUPLExx** field shows the COUPLE*xx* parmlib member used for system IPL.
- The **SMREBLD** field shows whether (value 1) or not (value 0) system-managed rebuild has been activated in the CFRM couple data set.
- The **Max SMlevel** field shows the maximum system-managed process level supported by this system.

For each couple data set you can enter one of the following command codes to initiate an action:

***A: Allocate alternate CDS**
Replaces the current alternate CDS for a selected CDS type with a new one. There are two options how to do this:

- The alternate CDS is allocated automatically by SA z/OS.

  This automatic allocation requires that spare volumes have been defined, and that one of these spare volumes is available. For details, see "SYSPLEX Policy Item" in *IBM Z System Automation Defining Automation Policy*.
- Specify the data set that is to be used as the new alternate CDS.

If you specify your own data set, observe the following:

- The data set must exist
- It must have been formatted with the XCF formatting tool
- It must be at least as large as the current primary CDS, which means that every value you have passed to the XCF formatting tool (for example, in the case of a sysplex CDS, the maximum number of systems supported) must be equal to or greater than the corresponding value of the primary CDS.

C: Display CHPIDs
Displays information about the channel paths for the selected CDS type.

D: Display CDS information
Displays detailed information about the selected CDS type. This comprises the formatting parameters and the policies that are contained in the CDS, if applicable. When the CDSs of the selected type contain policies, the detail information panel provides further actions:

D: Display policy
Displays details about the selected policy.

S: Start policy
Makes the selected policy the active policy. The policy switch must be confirmed before it is executed.

***P: Switch alternate CDS to primary CDS**
This action makes the alternate CDS the primary. Since an alternate CDS is no longer available after the switch, SA z/OS displays a confirmation panel before the action is performed. On this panel you can specify a new alternate CDS. When CDS recovery is switched on and you do not supply your own alternate CDS, SA z/OS tries to allocate a new alternate CDS automatically.The special requirements for manual and automatic creation of the new alternate CDS are the same as those for the replacement of the alternate CDS (action code A).

Making an Alternate CDS the Primary CDS

In this example, the alternate LOGR couple data set is made the new primary CDS. A new alternate CDS is automatically generated.

To switch the LOGR couple data set, enter P before L0GR on the initial INGPLEX CDS command dialog panel and press Enter. INGPLEX CDS displays a confirmation panel.

Use this panel to determine how a new alternate CDS is to be created after the switch. You can either specify your own new alternate CDS or let SA z/OS create it for you. When you specify the new alternate CDS yourself, the data set must exist and must have been formatted with the XCF formatting tool. Automatic creation requires that spare volumes have been defined for LOGR couple data sets.

Pressing PF10 causes SA z/OS to generate the new alternate CDS. After returning to the CDS command dialog, refreshing the panel, and scrolling down with PF8, the panel shows that the previous alternate LOGR CDS has become the primary, and there is a new alternate that was created by SA z/OS.

Switching the CFRM Policy

In this example, the active CFRM policy is switched.

Enter D before CFRM on the initial INGPLEX CDS command dialog panel and press Enter. The CFRM Couple Data Set Information panel is displayed.

The panel shows information about the names and locations of the CDSs. The panel also shows the parameters that were used by the formatting tool of XCF for the allocation of the CDS. The **POLICY** column, for example, displays the maximum number of policies the CDS can contain. The panel also shows information about the policies in the CDS, for example, how many coupling facilities and structures are defined in every policy, and which policy is currently active.

To switch to the HIRPOL policy, enter S before this policy and press Enter. INGPLEX CDS displays a confirmation panel.

Displaying the Channel Paths for a CDS Type

In this example, the channel paths for the CFRM couple data sets are displayed.

Enter C before CFRM on the initial INGPLEX CDS command dialog panel and press Enter. The CFRM Channel Path Information panel is displayed.

- The **System** column shows the name of the sysplex members.
- The **T** column (for 'type') indicates whether the CDS is the primary (value 'P') or alternate (value 'A').
- The **DEVN** displays the number of the device that the CDS resides on.
- The **CHPIDs** column shows the status of the paths to the devices in the format *chpid=status_code*. The codes are those of the operating system. They have the following meaning:

  +
  The path is logically and physically available and I/O on the path was successful.

  *****
  The path is physically, but not logically available. The subchannel's logical path indicator is off but I/O to the path is successful. You can use the command VARY PATH (*ddd*,*nn*),ONLINE to make channel path *nn* logically available to device *ddd*.

  –
  The path is neither logically nor physically available. The subchannel's logical and physical indicators are both off for this channel path. You can use the command CONFIG CHP(*nn*),ONLINE to make the channel path logically available to all devices connected to the channel.

  &
  The device is reserved to another path. This indicator applies to devices with the dynamic pathing selection feature.

  <
  The path is installed but not physically available. The start subchannel request received a condition code of 3.

  >
  The device microcode has detected an error and will not allow I/O to complete on the path.

  B
  The path is unable to communicate. The device indicates that a busy or reserve condition exists on the path.

  C
  A controller error occurred while accessing the device.

  D
  A device error occurred while accessing the device.

  I
  Intervention is required; the device is not ready.

  R
  The path is available and the device is reserved to this path/group. This only applies to devices with the dynamic pathing feature.

  T
  A time out has occurred; there is no response from the device. The cause of the time out is undetermined and this condition is transient.

  U
  A storage control unit or storage director error occurred while accessing the device.

  X
  Unable to determine the failing unit.
- The **SSID** field displays the storage subsystem that the device belongs to.

INGPLEX IPL

With the INGPLEX IPL function you can record, view and compare the IPL information of the operating system. If a system does not behave after IPL as expected, the IPL recording function enables you to identify parameters that were changed, for example, since the last IPL. The recording function enables you to compare different IPL scenarios. INGPLEX IPL is a tool that helps to identify and resolve the cause of startup problems. The following information can be displayed:

- The selected system (or blank)
- The name of the sysplex
- The maximum number of IPLs that are stored for each system
- An indicator showing whether comments in PARMLIB members are ignored when collecting information

The parameter DSN on invocation allows you to select a different IPL data set for displaying or comparing IPL information. It might be useful in case of a disaster recovery when you need IPL information of the sysplex that is down.

Use PF10 and F11 to scroll through all available columns on the INGPLEX IPL panel. SORT by column numbers is supported as well as the FIND and RFind command to locate information on the panel. You can also limit the display to a particular system by specifying the system name in the appropriate entry field.

The following command codes are available:

C
Compares the complete IPL information with another IPL record. A second panel will be displayed where you can select the second record.

D
Displays detailed information about this IPL record.

E
Erases the IPL information records. This action must be confirmed.

- The **Sysplex** field shows the name of the sysplex.
- The **System** column shows the name of the system in the sysplex.
- The **IPL Timestamp** column shows the date and time of the IPL. The format is YYYY-MM-DD HH:MM converted to local time zone.
- The **Dev** column shows the IPL device number.
- The **Volume** column shows the volume serial of the IPL device.
- The **OpSys** column shows the name of the operating system, for example, z/OS or OS/390.
- The **Release** column shows the release level of the operating system.
- The **FMID** column shows the FMID of the operating system.

For further information about the panel fields refer to the online help.

INGPLEX Dump Submenu

The dump functions can be invoked directly by specifying the commands, or from the dump panel of the INGPLEX command selecting the appropriate command. In addition, you can invoke the dump submenu from the main panel of the INGPLEX command selecting command **7**.

The dump functions that are available are:

1 INGPLEX SDump
This displays the default SDUMP options of all the systems in the sysplex. The dump options can be changed locally, sysplexwide, or for particular systems.

2 INGPLEX SVCdump
This displays the dump status information for each system in the sysplex. The function allows you to issue a multisystem dump of up to 15 address spaces for a single system, including their data spaces and structures.

3 INGPLEX SLIP
This displays SLIP (serviceability level indication processing) traps that have been set for all the systems in the sysplex. You can view, enable, disable, or delete each trap.

INGPLEX SDUMP

If you select option 1, the INGPLEX SDUMP panel is displayed.

The following command code is available:

C change
Invokes the modification panel by providing the options of the selected system as input

The fields on the command dialog panel display the following information:

- The **Sysplex** field shows the name of the sysplex.
- The **System** field shows the name of the system in the sysplex.
- The **Permission** field shows your authorization level.
- The **Dump options** field shows the default SDUMP options of all systems in the sysplex. For each system the following details are displayed:

  Q=
  Shows whether SDUMP quiesces the system while dumping the contents of the SQA or CSA.

  TYPE=
  Causes SVC dump to dump the cross memory address spaces that the caller has when SVC dump gets control (XMEM) or when the error causing the dump occurs (XMEME).

  BUFFERS=
  Shows the reserved storage exclusively used by SVC dump. This storage can be used when capturing the contents of the common area storage.

  MaxSpace
  Shows the maximum amount of virtual storage that SVC dump can use to capture volatile virtual storage data, summary dump data, and component-specific data before writing the dump to DASD.

  MsgTime
  Shows how long (mm) the message IEA793A is shown at the console. When the system deletes the message, it also deletes the captured dump.

The FIND and RFIND commands are supported.

If you specify command code C, a modification panel is displayed, which allows you to modify all SDUMP options. You can also delete SDUMP options. After entering your changes you can set the new options for:

- The selected system
- All systems in the sysplex
- Selected systems in the sysplex

To set the options press the appropriate F-key. If you want to modify selected systems in the sysplex, you are prompted for the systems that the SDUMP options are being changed on. To reset the options to the state when the modification panel was invoked press F5 Undo all.

Note: The user must be authorized to change any SDUMP option. The authorization can be any of those that are used for controlling coupling facilities and couple data sets.

For further information about the panel fields refer to the online help.

INGPLEX SVCDUMP

With the INGPLEX SVCDUMP command you can display the dump status information of each system in the sysplex. The INGPLEX SVCDUMP function allows you to issue a multisystem dump of up to 15 address spaces of a single system including their data spaces and structures.

On the INGPLEX SVCDUMP target system selection panel:

- The **Sel** field lets you select a system that a dump should be taken for.
- The **System** field shows the name of the system having joined the same XCF group the operator is logged on to.

For further information about the panel fields refer to the online help. After selecting a system and pressing Enter, the INGPLEX SVCDUMP Address Space Selection panel is displayed.

If you select the VTAM address space and the WATS address space, which is a user, press Enter and then PF5, the INGPLEX SVCDUMP Address Space Detail panel is displayed.

The VTAM address space has some data spaces (D), one list structure (L) and some XCF group members (M). TSO user WATS has nothing.

The following command codes are supported:

D
Deselects the previous selection.

S
Selects a local address space, data space, structure, or XCF group member address space for the SVC dump.

If you press PF5, the dump option selection panel is displayed.It shows the default dump options that are set on invocation. After specifying the dump title, press F5 to issue the dump. When the dump is taken, the function returns to the address space selection panel with all selections cleared.

The SORT, FIND and RFIND commands are supported for selection panels only. For further information about the panel fields refer to the online help.

INGPLEX SLIP

With the INGPLEX SLIP command you can display serviceability level indication processing (SLIP) traps being set at all systems in the sysplex. With INGPLEX SLIP you can view, enable, disable, and delete the SLIP trap defined in the sysplex.

The following command codes are available on this panel:

+
Shows the settings of the SLIP trap.

–
Hides the settings of the SLIP trap.

D
Disables the SLIP trap.

E
Enables the SLIP trap.

R
Deletes the SLIP trap.

The SORT, FIND and RFIND commands are supported.

Note: The user must be authorized to enable, disable, and delete a SLIP trap. The authorization can be any of those that are used for controlling coupling facilities and couple data sets.

For information about the panel fields refer to the online help.

How to Work with INGSTR

If you enter the INGSTR command without any parameters, a panel with all the coupling facilities of the sysplex is displayed.

The structure display shows all structures regardless of their allocation status or the coupling facility. You can limit the display by specifying a structure name pattern.

- The **P** column indicates whether a policy change is pending for the structure. Rebuilding the structure, if possible, will remove the pending status.
- The **D** column shows what type of duplexing is supported:

  U
  Indicates that user-managed duplexing is supported.

  S
  System-managed duplexing.

  B
  Both, where user-managed is preferred when possible.

  Note that, for performance reasons, this status does not include a check of the SMDUPLEX flag in the CFRM couple data set. However, this flag is checked when you use command code S.
- The **Old** column shows where the structure is allocated or where it was first allocated when it is being duplexed.
- The **New** column shows where the duplexed structure is allocated.
- The **Pref. Location(s)** column shows the locations where the structure should preferably be allocated. When the structure is allocated to the preferred coupling facility this column is blank. A '*' in front of the coupling facility name (or names) indicates that the structure does not allow XCF to perform a reallocation.
- The following command codes are available:

  D
  Display details of the selected structure.

  F
  Force the deletion of the selected structure.

  P
  Stop duplexing of the selected structure.

  R
  Rebuild the selected structure.

  S
  Start duplexing of the selected structure.

When the **Include unallocated** option is set to YES, all structures that are defined in the policy are shown. When the **Include condition** option is set to YES, the structure's current condition is also shown. Specifying this option increases the response time required to build the display.

Depending on the status of the CFs and the systems in the sysplex you can use the PF10 key to perform the XCF REALLOCATE command to move the allocated structures to their preferred location. When you press PF10 to move the structures, a panel asking you to confirm the action is displayed. Press PF10 to confirm the action, or PF11 to cancel the reallocation.

Note: You must be authorized to perform the FORCE, REBUILD, START, STOP, or REALLOC action.

Monitoring and Controlling Logical Partitions and Guest Systems

SA z/OS provides support for you to use Processor Operations (also known as ProcOps) to control and monitor:

- IBM Z and logical partitions, see “IBM Z and Logical Partitions” on page 180.

- Guest machines running under VM, see “VM Second Level Guest Systems” on page 183.

IBM Z and Logical Partitions

The IBM Z System Automation hardware interface function Processor Operations (ProcOps) offers a common set of hardware commands, that is, the ISQCCMD common commands. These commands allow you to manage and control the logical partitions (LPARs), the IBM Z central processor complex (CPC), the attached Support Element (SE), and the Hardware Management Consoles (HMCs) in the CPC's hardware network.

With ProcOps, you can invoke operator tasks normally done from an HMC workplace. You can use System Automation timers, automation table (AT) entries, or enter the commands on the same SA/NetView screen that you also use to operate the System Operations (SysOps) functions.

Since ProcOps commands are implemented as an application programming interface (API), you can build your own automation routines to handle individual IBM Z operations management use cases that are applicable to your environment. Read the following sections to find what commands to use for different use cases.

IBM Z Hardware Commands

These commands are executed on the SE, the CPC hardware itself, or both. The command completion details are returned to ProcOps in an asynchronous event. The overall duration of these hardware commands might vary and dependents on the machine or partition configuration.

- ACTIVATE - Activate a CPC hardware, a CPC image, or both.
- CBU - Activate the Capacity Backup record temporarily.
- DEACTIVATE - Deactivate a CPC hardware, a CPC image, or both.
- EXTINT - Issue a z/Architecture External Interrupt on a CPC image.
- CLRHWMSG - Remove a hardware message from the SE or HMC event view.
- LOAD - Load (IPL) an operating system or standalone program in a CPC image.
- OOCOD - Activate the Capacity on Demand record temporarily.
- RESTART - Perform a z/Architecture restart function on a CPC image.
- START - Perform the z/Architecture central processor start function.
- STOP - Perform the z/Architecture central processor stop function.
- STP - Manage the IBM Z Server Time Protocol (STP).
- SYSRESET - Perform the z/Architecture system reset function for a CPC image.
- TCM - Manage temporary capacity for different record types.

For a detailed description of these ISQCCMD common commands, see *IBM Z System Automation Operator's Commands*.

Special case: Sending an operating system command to a CPC image using the hardware console integration facility

In ProcOps, this IBM Z hardware command is implemented in the ISQSEND and ISQSNDH command APIs. For a detailed description of these host-based commands, see *IBM Z System Automation Operator's Commands*.

LPAR Runtime Weight and Capacity Management

You can use the ISQCCMD common command ICNTL to query and set the following information:

- The defined LPAR capacity
- The current LPAR minimum and maximum weights
- The LPAR weight capping Enabled flag
- The Workload Management Enabled flag
- The name of a Capacity Group Profile

For a detailed description of the ISQCCMD common command ICNTL, see *IBM Z System Automation Operator's Commands*.

Activation Profile Management

IBM Z activation profiles are permanently stored in the Support Element. You can use the ISQCCMD common command PROFILE to query and set values of the following CPC activation profiles:

- Reset profiles: Basic CPC configuration used at CPC activation.
- Image profiles: CPC image related configuration used at LPAR activation.
- Load profiles: Base information to load (IPL) an operating system in a CPC image.
- Group profiles: Membership of CPC images in a CPC resource or capacity management groups.

For a detailed description of the ISQCCMD common command PROFILE, see *IBM Z System Automation Operator's Commands*.

CPC and LPAR Information Queries

You can query the following CPC and LPAR information with the corresponding commands.

Queries	**Supported ISQCCMD Common Commands**
CPC Reset Activation Profile association	CCNTL
Consolidated CPC and LPAR details for all configured partitions	CPCDATA
Console (SE,HMC) network address and microcode level	CONDATA
CPC type, model, serial number, mode, status, or degraded mode details	CPCDATA, GETSINFO,GETSSTAT, GETSDGR
CPC Server Time Protocol (STP) network details	STPDATA
CPC temporary capacity record installation details	TCDATA
CPC image (LPAR) name, mode, status, active operating system type and version, current Program Status Word (PSW)	CPCDATA, GETIINFO, GETISTAT, GETIPSW
CPC image (LPAR) System Recovery Boost (SRB) status, boost efficiency indicators	GETIBOOST
CPC image (LPAR) last used LOAD address and current runtime token	GETILDI, GETITKN
CPC neighborhood information	GETCLUSTER

For a detailed description of these ISQCCMD common commands, see *IBM Z System Automation Operator's Commands*.

Miscellaneous IBM Z Hardware Commands

- RESERVE - Lock or unlock execution of disruptive commands on a CPC.
- CTRLCONS - Shut down or restart an IBM Z console (SE or HMC).

For a detailed description of the ISQCCMD common commands listed here, see *IBM Z System Automation Operator's Commands*.

IBM Z System Recovery Boost

In a System Automation environment, typical recovery situations are IPL and system shutdown. When a recovery situation is ongoing, normal workload processing stops. The objective of System Recovery Boost is to reduce the time to shut down, restart the system, process the backlog that accumulated during a system outage, and accelerate some sysplex recovery situations, by providing additional process capacity for the boosted image.

System Recovery Boost was introduced with IBM z15. On IBM Z subcapacity models, z/OS and other operating systems with this feature can temporarily run their shared general-purpose processors at full speed during boost periods. On IBM Z with zIIPs installed, z/OS images with shared zIIPs configured can run workloads that are designated for general-purpose processors, while in boost mode. Specific IBM Z temporary capacity records are offered to gain additional boost capabilities on demand.

Entering and leaving boost mode is managed on operating system level and includes activation procedures for the user. IBM terms and conditions regulate what boost periods are, their length, and usage frequency. Besides IPL and shutdown, IBM has grouped entitled boost mode use cases in recovery classes. For more information, see the white paper *System Recovery Boost for the IBM z15*.

System Recovery Boost Support in System Automation

Recovery Situations

At IPL time, a system can automatically participate in the boost unless the BOOST system parameter in IEASYSxx is set to NONE.

At shutdown time, System Automation can automatically activate System Recovery Boost upon shutdown of a system. For more information, see "Automatic Activation of System Recovery Boost Upon System Shutdown" in *IBM Z System Automation Customizing and Programming*.

Boost Status Monitoring

The target systems with an active SNMP connection to a ProcOps focal point are monitored if their LPARs are entering or leaving the boost mode. This monitoring is part of the standard connection polling. You can configure the poll frequency for a connection in the Processor Information policy of your policy database.

For target systems with the SYSCONS activated, the automation of z/OS messages indicating a boost mode change also update this status in ProcOps. To query the boost mode, you can use the **ISQXDST** Target Status Summary and Target System Summary panels or use the **ISQVARS GET** command with the target attention (TATTN) keyword. If the target system's LPAR is in boost mode, you can use the **ISQCCMD GETIBOOST** ***tgtname*** **OPT(STATUS)** to retrieve mode details.

The boost mode background monitoring by System Automation is useful because the IBM Z firmware itself does not emit an event if an image's boost mode change.

System Programmer Assistance

You can use the **ISQCCMD GETIBOOST OPT(EVALUATE)** command to collect IBM Z hardware information to evaluate if the CPC and LPARs are 'BOOST' capable and what is their ready state. The returned information includes the CPC's full or subcapacity indicator, the number of the physical installed zIIPs, the number of the shared zIIPs on LPAR level and capacity records IDs with zIIP capacity. Since

ProcOps has enterprise scope, you can run automated and scheduled queries without a need to log in to any Hardware Management Console (HMC).

VM Second Level Guest Systems

This feature provides processor operations support to control and monitor guest machines running under VM.

Processor operations allows an operating system to be IPLed into a processor, amongst other facilities. One such operating system is VM. Other operating systems can be IPLed within VM as guest machines. Of particular interest are LINUX guest machines, but MVS, VSE, and even VM guest machines may be possible. (Lower levels of guest machines are not considered). Thus SA z/OS offers an effective way to enter commands to and receive messages from such a guest target system in order to validate that it had IPLed correctly, or that it is behaving correctly.

With second level guest machine support you can:

- Capture messages issued by the guest machine itself and route these back to the ProcOps process for display or automated processing, or both
- Send commands to the guest machine from ProcOps, either as operator requests or automated actions

Processor Operations Dialog

The processor operations dialogs flow provides you with information about the guest systems hierarchy (see Figure 45 on page 183):

- Hosting systems are flagged on the Status Summary panel, ISQESUM.
- The ProcOps Service Machines (PSMs) of a hosting target system can be displayed by selecting the Target System Summary panel, ISQETARG, and the following PSM panel, ISQETRGG. The Target Hardware Summary panel for the PSM shows the initialized guest systems by PSM.
- You can display the hosting target system of a guest system by selecting the Target Hardware Summary panel, ISQEHARP.

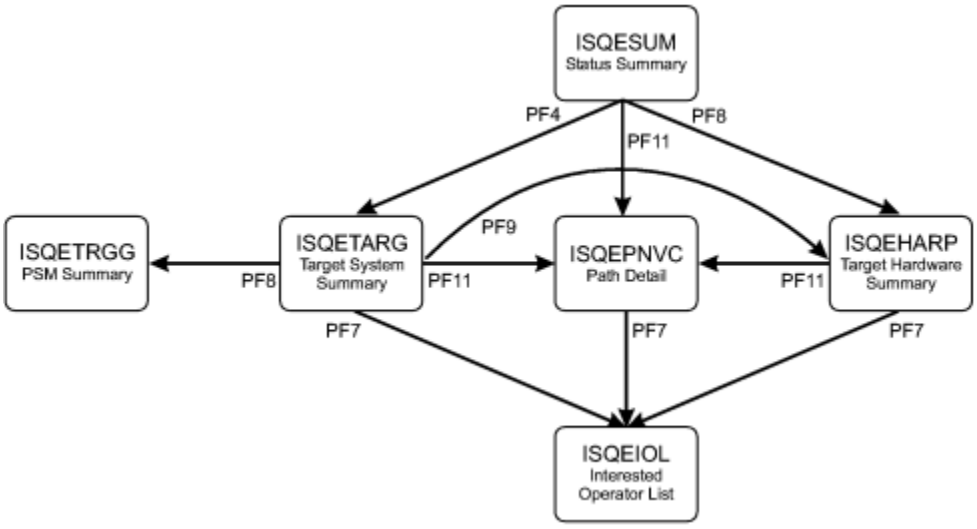


Figure 45. Processor Operations Panel Flow

Status Summary Panel (ISQESUM)

The Status Summary Panel, ISQESUM (Figure 46 on page 184), displays status information about processor operations target systems.

```
ISQESUM       SA z/OS  - Proc-Ops Target Status Summary    Updates: Dynamic
  Configuration in use: ACF

I isqxiii C isqxcls O isqxopt  A,B view netlogs E events   Debug Mode: OFF
S suspend /  H hold THW path       R Resume a suspended / hold THW path
Cmd    Target System   Status      Focal Points - Primary: IPSFO Backup: IPUFM
       BOEVMRAP  (*)   UNKNOWN
       CFA             CLOSED
       CFB             INITIALIZED
       CFF             INITIALIZED
       CF1             INITIALIZED
       CF61            INITIALIZED
       CF62            INITIALIZED
       KEYA            INITIALIZED
       KEYB            INITIALIZED
       KEYC            NOT ACTIVE
       KEYZ            NOT OPERATING
       KEY1            INITIALIZED
       KEY2            INITIALIZED
       KEY3            CLOSED
       KEY4            INITIALIZED
       KEY6            INITIALIZED
       KEY6B           UNKNOWN
       KEY7            NOT OPERATING
       LNXOLI1         CLOSED
       TEL01FE         CLOSED
       TEL17FE         UNKNOWN
       TEL30FE         UNKNOWN


Enter=Static  PF1=Help PF3=Exit PF4=Tgt Sys Summary PF5=Debug On/Off
PF6=Roll      PF7=Up   PF8=Down PF9=Tgt HW Summary  PF11=PATH Details PF12=Quit
```

Figure 46. Processor Operations Status Summary Panel (ISQESUM)

Targets that host other target systems are marked with (*), that is, they are *hosting systems*.

You can enter the command codes that are shown at the top of the panel in the **Cmd** field for a system to invoke related commands.

You can also use the PF keys that are given at the bottom of the panel to navigate to related panels.

Target Hardware Summary Panel (ISQEHARP)

Selecting the Target Hardware Summary panel, ISQEHARP, displays different fields depending on whether the target hardware is a ProcOps Service Machine (PSM) or not:

- Figure 47 on page 185
- Figure 48 on page 185

You can also use the PF keys that are given at the bottom of these panels to navigate to related panels.

```
ISQEHARP        SA z/OS Proc-Ops THW Summary               Updates: Dynamic

 Target Hardware Name: YORAMA     Type: PSM
 VM Host System Name:  VM123
 Target Hardware Description: ProcOps Service Machine for KEYn systems
 Initialized Target Systems:
  KEY6        KEY7         KEY8









 Communication Path Error Status:

 Last Significant Message:


 Enter=Static  PF1=Help PF3=Exit PF6=Roll PF7=Operlst  PF11=PATH Dtls PF12=Quit
```

Figure 47. Processor Operations Target Hardware Summary Panel (ISQEHARP) for PSMs

In Figure 47 on page 185 PSM in the **Type** field indicates that the processor is a ProcOps Service Machine (PSM) and that the initialized target systems are guest systems. For PSMs this panel :

- Displays **VM Host System Name**
- Does not display the **Mode**, **Model**, or **Serial Number** fields

For real hardware, the panel (Figure 48 on page 185) displays:

- The **Mode**, **Model**, and **Serial Number** fields
- Initialized Target Systems that are hosting target systems are flagged with (*)

```
ISQEHARP        SA z/OS Proc-Ops THW Summary               Updates: Dynamic

Target Hardware Name: YORAMA     Type: Mainframe   Mode: LPAR
                                 Model: n/a        Serial Number: n/a
Target Hardware Description: Host for VM systems
Initialized Target Systems:
 SYS1    (*) SYS2         SYS3









 Communication Path Error Status:

 Last Significant Message:


 Enter=Static  PF1=Help PF3=Exit PF6=Roll PF7=Operlst  PF11=PATH Dtls PF12=Quit
```

Figure 48. Processor Operations Target Hardware Summary Panel (ISQEHARP) for Non-PSMs

Target System Summary Panel (ISQETARG)

Selecting the Target System Summary panel, ISQETARG, displays a panel similar to Figure 49 on page 186.

```
ISQETARG          SA z/OS Proc-Ops Target System Summary     Updates: Dynamic

Target System
  Name         : KEY6____ (*)
  Description  :
  O.S.         : MVS
  Load Profile :
  Status       : INITIALIZED
  Attention    : EXCEPTIONS

Target Hardware
  Name         : YORAMA       Mode :  LPAR
  Attention    :
  Path Status  : ACTIVE

Target Hardware LPAR
  Name         : KEY6         Mode :  ESA
  Image Profile: KEY6

Last Significant Message:


Enter=Static  PF1=Help PF3=Exit  PF6=Roll PF7=Oper List  PF8=Guests
PF9=Target Hardware    PF11=Path Detail   PF12=Quit
```

Figure 49. Processor Operations Target System Summary Panel (ISQETARG)

If the target system is a hosting system:

- (*) after the **Name** field indicates that this is a hosting system
- PF8=Guests is displayed: Pressing PF8 displays the ISQETRGG panel.

Guests Summary Panel (ISQETRGG)

Selecting the Guests Summary panel, ISQETRGG, displays a panel similar to Figure 50 on page 186.

```
ISQETRGG       SA z/OS Proc-Ops PSMs/Guests Summary         Updates: Dynamic

VM Host System Name:

Defined ProcOps Service Machines:
 PSM01      PSM02

Initialized Target Systems:
 PSM01/Linux1     PSM01/Linux2     PSM01/Linux3     PSM02/MVS1
 PSM02/VSE1

⋮

Enter=Static  PF1=Help PF3=Exit PF6=Roll PF7=Up PF8=Down PF12=Quit
```

Figure 50. Processor Operations PSMs/Guests Summary Panel (ISQETRGG)

This panel displays the guest (that is, hosted) systems for a target system that is a hosting system. The guest systems are displayed for each associated PSM with the following:

VM Host System Name
Displays the name of the hosting system.

Defined ProcOps Service Machines
Lists PSMs defined in the control file to run on the hosting system.

The number of PSMs per hosting VM system is limited to 12. (This is checked by the configuration dialog.)

Initialized Target Systems
Lists target systems currently associated with the hosting system. The target systems are identified by PSM/Target.

The number of guest systems being controlled by a PSM is limited to 60. (This is checked by the configuration dialog.) Therefore the total number of guest systems running on a VM host is limited to 720.

PATH Detail Panel (ISQEPNVC)

Selecting the PATH Detail panel, ISQEPNVC, displays a panel similar to Figure 51 on page 187.

```
ISQEPNVC             SA z/OS Proc-Ops  PATH Detail         Updates: Dynamic

PSM Name            : SAFOS                 Connection Type     : TCPIP

Message Server Port : 4711_                 Command Server Port : 4712_
IP Stack            : STACK---
PSM IP Address or Hostname:
999.999.999.999


Security Resource Name  : DEIBMIP1.IP3T1200
Connection Authorization: PROCOPS           Lock Holder    :

PATH Status    :

Poll Frequency : 40
Poll Retries   : 2

Last Significant Message:



Enter=Static           PF1=Help          PF3=Exit          PF6=Roll
PF7=Oper List          PF12=Quit
```

Figure 51. Processor Operations PATH Detail Panel (ISQEPNVC)

For PSM hardware TCPIP address, Message Server port, Command Server port, IP stack, and Security resource name are displayed.

Security Resource Name
Identifier used for SAF checking (for example, by RACF®).

PSM Console Mode

Normally the PSM runs disconnected. In this mode, its regular, unsolicited messages are not seen, even though they are recorded by the Logger thread in its log data set. To assist problem determination, these regular messages are captured by the Message Handler, queued to the Message Queue, and routed to the current ProcOps focal point by the Message Server. They should then be seen in the NetView log.

When a user that is logged on to the PSM issues a PSM request, a CP command or a CMS command, the responses are always displayed to the user. So, if a user logs on to the PSM no unsolicited messages are seen. These unsolicited messages can be made visible by issuing the following command on the VM console:

```
CP SET VMCONIO OFF
```

If the Message Handler is restarted for any reason (for example, when the first guest machine on the host is initialized using the ISQXIII command), the unsolicited messages to the VM console may be again lost. Reissue the CP SET command to restore message display at the VM console.

If an ISQTRACE command is issued that sets any thread trace to the ON state, the messages issued by the PSM are automatically displayed at the VM console, and not sent to NetView. This is to avoid possible message loops, because each message sent may cause additional messages to be created.

Note that if indentation of messages is active, the PSM messages returned to NetView also appear indented. If any PSM message spills to another line, that new line is a separate message sent to NetView.

If an ISQTRACE command is issued, and all traces are set to the OFF state, the flow of unsolicited messages is once again routed to NetView. This means that the effect of any earlier `CP SET VMCONIO OFF` command is then lost.

When messages are displayed on the VM console, they may also be spooled to a VM output file or a reader file. This includes any trace messages.

When a user is logged on to the PSM virtual machine, the following points should be noted.

- If messages are sent to the VM console they may cause the PSM to halt execution temporarily when or shortly after the screen fills with messages. Consider entering the CP TERMINAL MORE command to control the effect of this (possibly as a command in an initial EXEC that invokes the PSM Control Program). For example, issuing CP TERMINAL MORE 0 0 means that the VM console never locks up when regular messages are issued.
- Some messages are highlighted and held. For example, the XAUTOLOG command causes such messages. When the screen fills (even if CP TERMINAL MORE has been issued) the session goes into HOLDING mode and requires operator intervention.
- Any CP command can be executed by prefixing it with CP. Any CMS command can be executed by prefixing it with CMS. However any command is executed by the Command Server thread and serializes that thread. If such a command requires an extended period of time to execute it can prevent the Command Server thread from responding to requests. If a full screen command (such as XEDIT) is entered, all threads of the PSM Control Program stop executing until the user leaves the full screen command.

Starting the PSM Control Program

The PSM Control Program runs in its own virtual machine. To start the PSM Control Program manually, logon to the virtual machine in the usual way. The PSM Control Program can be started by simply entering its program name, as selected by the programmer at installation time. For example, if the programmer named the program ISQPSM, simply enter this in response to a Ready prompt:

```
Ready; T=0.04/0.05 19:52:14
isqpsm
ISQMA0001I PSM Main Thread entered
ISQMA0101I Thread 3 started as CSERV
ISQMA0002I Waiting for Command Server thread to initialize
 ISQCS0710I Command Server thread entered
 ISQCS0205I Command Server valid FP addresses are 9.164.172.63 9.164.156.189 9.99.99.99 66.77
ISQMA0101I Thread 4 started as CNSERVER
ISQMA0101I Thread 5 started as LOGGER
 ISQCN0710I Console Server thread entered
 ISQCN0711I Console Server thread initialized
 ISQCS0711I Command Server thread initialized     ISQLG0710I Logger thread entered
 ISQLG0711I Logger thread initialized
```

Initially PSM messages are displayed on the VM console. These may be captured to a disk file by spooling the console file to a reader file. To initiate spooling enter a CMS SPOOL command; for example:

```
spool console start *
```

Then start the PSM, as above. When you wish to create a complete file in the reader, enter a command:

```
spool console close
```

Spooling continues, and you may use the above spool console close command to create successive reader files. When spooling is to stop, enter the command:

```
spool console stop
```

(This does *not* write the remaining records to a reader file.)

If the above commands are to be entered while the PSM Control Program is running, the *spool* commands should be prefixed with *cms*:

```
cms spool console close
```

To automate the process of starting the PSM Control Program when the PSM virtual machine starts, add the ISQPSM command to the PROFILE EXEC of the PSM virtual machine.

On occasions, you may wish to start the PSM Control Program with traces active for threads that are started automatically. Before issuing the command to start the PSM, enter a CMS command of the following form:

```
GLOBALV  ISQTrace.comp   ON
```

Where *comp* is the trace option as used in the ISQTRACE request. (Usually one of CSERV, TCPIP, LOGGER or CNSERVER). Enter one command for each trace required.

Stopping the PSM Control Program

To stop the PSM Control Program normally, first logon to the PSM virtual machine. Because the PSM virtual machine runs disconnected, the first action is to issue a BEGIN command to resume operation.

```
    LOGON PSM
     ICH70001I PSM  LAST ACCESS AT 17:08:54 ON MONDAY, OCTOBER 20, 2003
    z/VM Version 4 Release 3.0, Service Level 0202 (64-bit),
    built on IBM Virtualization Technology
    There is no logmsg data
    FILES: 0038 RDR, 0003 PRT,   NO PUN
    RECONNECTED AT 20:25:10 CST WEDNESDAY 10/20/03

     begin
```

Then enter the STOPALL request. The PSM control program should then close its threads.

After a short period of time (about ten seconds) the CMS prompt should be given:

```
    stopall
     ISQCN0402I Command <stopall> entered
    HCPMFS057I PSM not receiving; not authorized
    ISQCS0218I Request <stopall> done.
     ISQCN0708I Console Server thread canceled
     ISQCN0709I Console Server thread ending...
     ISQCS0708I Command Server thread canceled
     ISQLG0708I Logger thread canceled
     ISQCS0709I Command Server thread ending...
     ISQLG0709I Logger thread ending...
    ISQMA0005I Main Thread ending (others may still be running)
    Ready; T=0.09/0.10 20:30:59
```

Sometimes it is not possible to use STOPALL to stop the PSM. In these cases, enter the HX command (halt execution) followed by B (BEGIN), repeatedly until the Ready prompt is obtained:

```
    hx
    CMS
    b
    DMSRXS1419E EventSignal failed for event CONNSOCK; RC=8 Reason=108
    Ready; T=0.14/0.14 20:36:31
```

The DMSRXS1419E message is normal

Restarting the PSM Control Program

After a PSM Control Program has stopped control returns to CMS and a CMS Ready prompt is issued to a logged on user. At this point it is advisable to enter a HX command to terminate any running threads.

The PSM Control Program can be restarted by entering its program name, for example, ISQPSM.

When the program restarts any trace setting that were set from the last execution are retained and are still in effect. If you wish to set off a trace before resuming execution enter the CMS command:

```
GLOBALV  ISQTrace.comp   OFF
```

Where *comp* is the trace options as used in the ISQTRACE request. Enter such a command for each trace that is to be set off.

How SA z/OS Uses Error Thresholds

Error thresholds influence whether SA z/OS recovers from an error situation.

For applications, you can define a critical threshold for restarting these applications. This is a number of error conditions within a certain time interval, for example, five error conditions requiring restart within one hour. During a condition requiring restart, SA z/OS checks whether the number of occurrences of the condition reaches the critical threshold. If it is reached, SA z/OS does not attempt to restart a resource.

For z/OS components, such as dump data sets or log data sets, you can define thresholds to limit the frequency of how often they may be deleted after they have filled up without an action being taken or a notification being sent to the operator.

Error thresholds also determine when you should be alerted to problems. The primary use of error thresholds is to track subsystem abends and ensure that the abend and restart cycle does not become an infinite loop, but they may also be customized for other uses.

Refer to *IBM Z System Automation Defining Automation Policy* for information on how to define error thresholds. The following sections describe how to obtain information about them.

Displaying Threshold Settings and Occurrences

SA z/OS counts the number of times that a subsystem abends. Each time that an abending subsystem is restarted, a check is made to see whether the restart is made too often.

SA z/OS allows you to specify thresholds in terms of error rates. If the measured error rate reaches the *critical* threshold, SA z/OS stops to recover the resource and posts it to a broken status.

You have the following options to obtain information about threshold settings and to view when thresholds have been reached:

INGTHRES
If you want to display all defined thresholds or add, change, or delete threshold settings for a particular resource, use the INGTHRES command. This displays the related Command Dialogs panel.

DISPERRS
If you want to check the error counts for resources on a specific system, issue a DISPERRS command to this target system. SA z/OS displays the DISPERRS Command Dialogs panel with a list of all the errors that have been recorded for resources on the target system.

Table 17 on page 190 provides details of how to use commands to display threshold settings and occurrences.

Table 17. Displaying Threshold Settings and Occurrences

Task	Details
Display Threshold Settings for a Selected Resource: INGTHRES	Enter INGTHRES on any command line to display a panel that shows the settings for the critical, frequent, and infrequent thresholds for the specified resource.

Table 17. Displaying Threshold Settings and Occurrences (continued)

Task	Details
Display Occurred Errors: DISPERRS	If you specify the DISPERRS command, you get the DISPERRS command dialog, which displays the number of errors for all the applications on the target system and also when an application has reached its threshold. Scroll through the panel to find the resource that has failed. If a subsystem is in the BROKEN status and it has not exceeded its critical threshold, it has probably encountered an unrecoverable error. If you view either SDF or DISPINFO for it, you should find the text of the AOF571I status change notification message that explains why the subsystem status was posted as BROKEN.
Display Occurred Errors for a Specific Resource: DISPASF	Although DISPERRS displays the errors that have occurred for all of a system's resources, DISPASF lets you see detailed information about errors for a resource on a specific system. For example, if you want to see more information about the error with resource MVSDUMP on system AOC7 from the DISPERRS command dialog panel, issue the following command: `DISPASF MVSDUMP TARGET=AOC7`

How to Set or Change Error Thresholds

About this task

You can set or change thresholds for a resource, the resource group defaults, or the system defaults using the INGTHRES command dialog.

Procedure

1. Enter `INGTHRES` *`resname`* at a command line to display the INGTHRES panel with the settings of the critical, frequent, and infrequent thresholds for the specified resource.
2. Use the following command codes against the resource to change the Critical, Frequent, or Infrequent thresholds:

   A
   Add thresholds

   C
   Change thresholds

   D
   Delete thresholds

   If you are adding or changing a threshold, type the number of errors and specify the time period. Press Enter to set the threshold values. If you are deleting the thresholds, press Enter to set the values to null, which deletes them.

How to Reset Occurrences

About this task

You can reset occurrences of threshold errors by updating records in the automation status file. To do this enter the following command:

```
ASF ID=resource_name REQ=REPL ERRORDT='' THRSHLD=''
```

Where *resource_name* is the name of the resource that you want to reset the occurrence data for. Message AOF001I informs you that the update was completed:

```
AOF001I REQUEST "REPLACE" WAS SUCCESSFUL FOR "resource_name"
```

Setting Up Thresholds Using the Customization Dialog

SA z/OS uses error thresholds to determine when to stop recovering from an error situation. You can define error thresholds to track errors for MVS components and applications.

You can set error thresholds for different policy objects using either the THRESHOLDS or MINOR RESOURCES policy item (for details, see *IBM Z System Automation Defining Automation Policy*).

To set error thresholds for all MVSCOMP resources, use the THRESHOLDS policy item of the MVSCOMP Defaults policy object. To set error thresholds for all applications, use the THRESHOLDS policy item of the Application Defaults policy object. If you do not specify error threshold defaults for MVS components or applications, SA z/OS uses the error thresholds for all monitored resources that are defined in the System Defaults policy object.

You define thresholds by specifying the number of errors in a particular time interval, for example three times in one hour.

Note: Setting a *critical threshold* to 1 disables the recovery process at the first occurrence of the error. This is valid independent of any time interval that is specified.

When the number of occurrences of a situation reaches the infrequent and frequent error thresholds, SA z/OS notifies the operator and logs a message in the NetView log. When the number of occurrences of a situation reaches the critical threshold, SA z/OS stops the recovery from that abend situation.

Once processing has stopped, operator intervention is required to restart it. This is because the critical error threshold is used as a trigger to stop SA z/OS from attempting to restart the application. After the operator has restarted the application, the thresholds will next be analyzed when the application next abnormally ends. If this occurs within the time interval specified for the critical threshold, it may result in the critical threshold being reached again. The error density will decrease with time, but this will not cause SA z/OS to restart the application.

For example, suppose an application has a critical error threshold of two per hour:

- The application will be restarted if it abnormally ends at 10:30.
- The application will be restarted if it abnormally ends again at 12:20.
- The application will not be restarted by SA z/OS if it abnormally ends again at 12:45. Operator intervention is required to restart the application.
- If an operator restarts the application at 13:05 and the application abnormally ends at 13:15, the error count will again reach two per hour and the application will not be restarted by SA z/OS.
- If the operator restarts the application at 15:00, the error count within the last hour will be zero again. The application will have to abnormally end twice within the hour in order to reach the critical error threshold again.

The primary use of error thresholds in SA z/OS automated operations is to track application abnormal ends and ensure that the abend-and-restart cycle does not continue indefinitely. You can also set error thresholds for events other than the occurrence of errors. For example, you can set thresholds to act as counters of specified events on your system.

Thresholds for Applications

Error threshold definitions for an application define how many abend errors can occur before a message is logged or restart processing is stopped.

You define such a threshold using the THRESHOLDS policy item of an Application policy object. SA z/OS displays the Thresholds Definition panel.

If no error thresholds are set here, the application uses the thresholds that are defined in the Application Defaults policy object. Determine whether existing default threshold values are appropriate for this application. If they are not appropriate, create a unique threshold entry for the application.

Thresholds for Minor Resources

You can define thresholds for minor resources (such as messages or statuses) that can trigger automation for applications and MVSESA resources.

You can do this using the MINOR RESOURCES policy item of an APL or MVC policy object.

For example, a threshold for MVSESA.MVSDUMP specifies how often a full dump data set should be recovered in a particular time interval before the operator gets a message indicating that something has gone permanently wrong.

Thresholds can be set that let the operator know if certain errors are occurring infrequently, frequently or have reached a critical stage where the recovery process should be ended (to avoid endless loops). This is done by specifying how many times an error must happen in a certain time period for each error situation.

Handling Jobs That Are Not Controlled by SA z/OS

The INGLKUP command dialog displays all active jobs, started tasks (STC), and APPC/MVS transaction programs that are not controlled by SA z/OS. You can either stop, cancel, or force these jobs. Refer to the *IBM Z System Automation Customizing and Programming* for full details.

For each job the following information is displayed:

- Job name
- Step name
- Procedure name
- Job type
- Address space ID (this is only shown when the job name is not unique)

Communicating with Other Systems

This chapter discusses the following topics:

Sending Commands within a Sysplex

This section provides details of the prerequisites for communicating with systems in the following environments:

- Within a sysplex: this is called a *single system image*.
- Within your complete enterprise: this is called a *single point of control for the enterprise*.

There is also a series of commands that is directly processed by the automation manager of a sysplex. You do not need a target parameter for these commands. You can retrieve sysplexwide information, such as the statuses of resources within the sysplex. For a complete list of sysplexwide commands, refer to the *IBM Z System Automation Customizing and Programming*.

The two environments allow you operate differently:

Single system image
You can use a system within the sysplex to monitor and manipulate all other systems in the sysplex. Then each system in the sysplex looks like the one you are currently working on.

Single point of control
If your system is the focal point system, you can use it to manipulate all the resources of your enterprise from this focal point system by issuing NetView commands. The target systems do not need to be members of your sysplex.

To exploit these features, the following prerequisites and conditions need to be fulfilled:

- **Single system image:** The target system is within the local sysplex.
- **Single point of control:** The target system has a direct gateway connection with the local system.

The following prerequisites also need to be fulfilled for both cases:

- The target is active.
- SA z/OS on the target has been fully initialized.
- The target can be reached either by XCF or RMTCMD.

SA z/OS searches your environment for the target system in the following order:

1. System name within the local sysplex
2. Domain ID within the local sysplex
3. Local sysplex name
4. Domain ID within the enterprise
5. System name within the enterprise
6. Sysplex name within the enterprise

You have several choices as to how to route commands to other systems within the sysplex:

- "INGEXEC" on page 194
- "INGSEND" on page 194
- "Single System Image and Single Point of Control" on page 194
- "Cross-Domain Sessions" on page 195

INGEXEC

You can use the INGEXEC command as a system utility to process a specified command on the system, or systems, where the specified resource or resources reside without having to specify the systems. The INGEXEC command interrogates the automation manager to determine the list of resources that are affected. The INGEXEC command operates sysplex-wide.

For more information about the INGEXEC command, see *IBM Z System Automation Programmer's Reference*.

INGSEND

This will assist you in sending a command over a semi-shared RMTCMD session (this requires some configuration).

1. Enter INGSEND without parameters at the command line of any NetView or system operations panel to display the INGSEND panel.
2. Complete the other fields and press Enter. (For more information on the individual fields, refer to *IBM Z System Automation Customizing and Programming*).

Single System Image and Single Point of Control

If the command you want to send is an SA z/OS command and is supported by single system image (most of them are), you can just issue it, specifying the destination system as the target. This only requires that NetView is active on both systems: VTAM is not used.

The following example shows the usage of single system image to determine the status of APPL1 on every system within the sysplex (AOCA, AOCB, AOCC, AOCD) and then to restart it on AOCC where it is BROKEN. All operations are carried out from the IPUFA system.

```
  DISPSTAT APPL1,TARGET=*ALL
```

```
CMD  RESOURCE     STATUS    SYSTEM   JOB NAME  A I S R T RS TYPE     Activity
---  -----------  --------  -------- --------  ------------ -------- ---------
 _   APPL1        AUTODOWN  AOCA     APPL1     Y Y Y Y Y Y  TRANS    --none--
 _   APPL1        DOWN      AOCB     APPL1     Y Y Y Y Y Y  TRANS    --none--
 b   APPL1        BROKEN    AOCC     APPL1     Y Y Y Y Y Y  TRANS    --none--
 _   APPL1        CTLDOWN   AOCD     APPL1     Y Y Y Y Y Y  TRANS    --none--
```

Type B against APPL1 on AOCC and press Enter.

Complete the command parameters (STATE=RESTART, START=YES) and press Enter to issue the command. It will be sent to the target system and executed.

Cross-Domain Sessions

Cross-domain sessions let you send commands to other domains using NetView RMTCMD, gateways, or Terminal Access Facility (TAF).

To display cross domain sessions, from the NetView command line, type A0C to invoke the SA z/OS main menu.

From the SA z/OS main menu, type **1** (Operator Interface) and press the Enter key. This takes you to the Main Operator Menu. At the command line, type **1** (CROSS DOMAIN) and press the Enter key. This leads to the Cross Domain Menu. From this menu you can display cross-domain sessions.

Sending Commands to a Linux Target System

Use the ISQSEND command of processor operations to send a command to a Linux target system.

The system previously must have been initialized successfully, using the ISQXIII command. In order to see the responses on the NetView screen, the command ISQXMON must have been used to add you to the interested operator list of the target system.

Since the Linux operating system handles commands case sensitive, you have to use the "netvasis" prefix. In the following example, the Linux command `ps -x` is issued to show Linux process related information. Note, that NetView suppresses the "netvasis" prefix when redisplaying the entered command.

```
NCCF                     Tivoli Netview   IPVFJ TIL      14/28/01 15:19:09
 * IPVFJ    ISQSEND FCLX2 OC PS -X
 U IPVFJ    ISQ901I  FCLX2 SC ISQ417I CMD STATUS(ACCEPTED)
 U IPVFJ    ISQ901I FCLX2    OC  PS -X
 U IPVFJ    ISQ017I ISQSEND COMPLETED SUCCESSFULLY.
 U IPVFJ    ISQ901I FCLX2    OC    PID TTY      STAT   TIME COMMAND
 U IPVFJ    ISQ901I FCLX2    OC      1 ?        S      0:00 init
 U IPVFJ    ISQ901I FCLX2    OC      2 ?        SW     0:00 kMCHECK!
 U IPVFJ    ISQ901I FCLX2    OC      3 ?        SW     0:00 keventd!
 U IPVFJ    ISQ901I FCLX2    OC      4 ?        SW     0:00 kswapd!
 U IPVFJ    ISQ901I FCLX2    OC      5 ?        SW     0:00 kreclaimd!
 U IPVFJ    ISQ901I FCLX2    OC      6 ?        SW     0:00 bdflush!
 U IPVFJ    ISQ901I FCLX2    OC      7 ?        SW     0:00 kupdated!
 U IPVFJ    ISQ901I FCLX2    OC    306 ?        SW     0:00 keventd!
 U IPVFJ    ISQ901I FCLX2    OC    309 ?        SW     0:00 keventd!
 U IPVFJ    ISQ901I FCLX2    OC    312 ?        SW     0:00 keventd!
 U IPVFJ    ISQ901I FCLX2    OC    315 ?        SW     0:00 keventd!
 U IPVFJ    ISQ901I FCLX2    OC    436 ?        SN     0:00 syslogd -M 0
 U IPVFJ    ISQ901I FCLX2    OC    450 ?        SN     0:00 klogd
 U IPVFJ    ISQ901I FCLX2    OC    469 ?        SN     0:00 crond
 U IPVFJ    ISQ901I FCLX2    OC    488 ?        SN     0:00 inetd
 U IPVFJ    ISQ901I FCLX2    OC    507 ?        SN     0:00 httpD
 U IPVFJ    ISQ901I FCLX2    OC    554 console  S      0:00 -bash
 U IPVFJ    ISQ901I FCLX2    OC    603 console  R      0:00 ps -X
 U IPVFJ    ISQ901I FCLX2    OC  root@boelnfc2 /root!#
 ------------------------------------------------------------------------



 ???
```

Figure 52. Sending a Command to a Linux Target System

Terminal Access Facility Full-Screen Sessions

The NetView terminal access facility (TAF) allows you to log on to multiple applications either on your system or other systems.

IBM Z NetView Command Reference Volume 2 (O-Z) describes in detail how to use TAF.

SA z/OS allows you to define TAF sessions in the customization dialog so that you do not have to set them up each time you want to use them. Refer to *IBM Z System Automation Defining Automation Policy* for more details.

Determining the Status of TAF Full-Screen Sessions

About this task

Using the TAF Fullscreen Menu panel you can see the list of TAF fullscreen sessions that have been defined to SA z/OS for your operator ID, and determine which of these are active.

To see the list of TAF fullscreen sessions available to you:

Procedure

1. From the NetView command line, type **Oper**. You see the Main Operator Menu.
2. On the Selection line, type **1** (CROSS DOMAIN) and press the Enter key. You see the Cross Domain Menu.
3. On the Action line, type **2** and press the Enter key. You see the TAF Fullscreen Menu.

   Note: If no TAF sessions are defined for your operator ID, you will see a line mode message

   ```
   AOF041I UNABLE TO FIND FULL SESSION ENTRIES
   ```

```
AOFK2DF                   SA z/OS  - Command Dialogs      Line
Domain ID   = IPUFM     --- TAF Fullscreen Display ---    Date = 14/11/04
Operator ID = OPER1           System  = AOC7              Time = 16:34:53

Cmd:  I Initialize     R Return to    T Terminate

Cmd Name      Application  System    Status      SRCLU     Session id
--- --------  -----------  --------  ----------  --------  ----------
 _  TSO       IPSFM        KEY3      ACTIVE      TAFFJF35
 _  IMS       IPSFN        KEY4      INACTIVE

⋮

Command ===>
  PF1=Help     PF2=End        PF3=Return                        PF6=Roll
                              PF9=Refresh                      PF12=Retrieve
```

Figure 53. TAF Fullscreen Display Panel

This panel shows the settings of all fullscreen sessions that are defined in the policy database.

Managing TAF Full-Screen Sessions

You can use the *TAF Fullscreen Display* panel to log on to other applications on the current or another system to return to the session or to terminate the session.

For each TAF session the following information is displayed:

Name
Is the name of the session.

Application
Is the name of the application as defined in the VTAMLST data set.

System
Is the name of the system where the application runs. This name is defined in the policy database.

Status
Is the current status of the TAF session.

SRCLU
Is the logical unit name of the session.

Session ID
Is the session ID as assigned by NetView.

You can use the following command codes:

I
Initializes the fullscreen session.

R
Returns to session.

T
Terminates the fullscreen session.

Gateway Sessions

SA z/OS does not inform an operator through a message when an outbound session with a remote system is established or not. Therefore, you need to monitor session status yourself.

In SA z/OS you can monitor other systems and issue commands to those systems using gateway sessions.

Gateways are useful because they:

- Save you the step of logging on to the NetView on the other system
- Let you communicate with another system even if you do not have an ID for that system
- Can provide a second NetView session with a system if you have an ID for that system

SA z/OS provides several panels that let you check your communication links with other systems and send commands to those systems. You can:

- Determine the status of gateway sessions
- Determine the status of paths. A path uses an intermediate system to connect two systems that are not directly linked
- Issue commands to other systems using gateway sessions

Determining the Status of Gateway Sessions

You can determine which gateway sessions have been defined and see which of those are active. You may want to do this if you use SDF to monitor other systems, since SDF uses gateway sessions to receive information from remote systems.

You also need to check the status of a gateway session if you want to receive notification messages from other systems. (Notification messages are messages that document significant actions that SA z/OS has detected or taken.) Finally, you need to check the status of a gateway session if you want to issue a command to another system using a gateway session.

You can obtain information on gateway sessions from the Gateway Display panel or from SDF, if SDF has been customized to do this.

To access the Gateway Display panel, type DISPGW from any command line and press Enter.

For further information on the DISPGW command, see *IBM Z System Automation Operator's Commands*.

Chapter 9. Solving Problems with Resources

For most operators, the key resource to be monitored is the application. You want to be sure that your enterprise's critical jobs and subsystems are getting the service they require and are operating satisfactorily.

This information describes general steps that you can take if resources do not start or stop as expected. It also contains a scenario that shows how you can use SA z/OS commands to discover and react to problems affecting your important applications.

Analyzing Problems with Resources

If a resource is not in a status as you expect, you can start the analysis by using the INGWHY command.

INGWHY is an operator support function that helps operators to initially analyze situations to find why automation took a resource into the displayed status, or why automation was unable to take a resource into the desired status. This information is valuable when operators open trouble tickets or when automation administrators or system programmers need further in-depth investigations. Or, for simple situations, INGWHY helps operators to fix the problem themselves without contacting the subject matter experts.

INGWHY interrogates the automation manager and the various agents for dependency, status and other information, and based on that data it evaluates potential reasons and offers next step actions.

Actions need special considerations. Actions might differ from company to company or even from resource to resource, but INGWHY allows the automation administrators to adapt the proposed actions to their needs. A DSIPARM member INGWHYU is provided for user-defined actions. For more information, see "Defining INGWHY User Actions" in *IBM Z System Automation Customizing and Programming*.

If Resources Do Not Start

If you have attempt to start a subsystem and SA z/OS does not do it, there are some things you can check:

1. Use INGWHY command to initially analyze the situation to see possible reasons for the current state of the resource. For more details, see *IBM Z System Automation Operator's Commands*.
2. If INGWHY doesn't help, locate and examine the AOF313I message for the resource. This should clearly state why the startup request was rejected.
3. If there isn't a AOF313I message, search your netlog for more information.
4. Use the INGLKUP *resource* REQ=COLLECT command to collect diagnostic information about the resource from several SA z/OS commands (such as INGAMS, INGINFO, DISPMTR, DISPSTAT, and so on). You can also specify a data set for the output of the INGLKUP command. If you do not specify a data set, the output is written to the netlog. For more details, see *IBM Z System Automation Operator's Commands*.

Other issues to consider include:

- **Supporting resources are not up**

  If supporting resources are not up, SA z/OS tries to start them. If you wait for a short period of time they should come up and then the target subsystem will be started.

  You can check on the state of the parents through the DISPTREE display, including status, or through messages issued to the Network Communications Control Facility (NCCF) console. If for example, it turns out that a parent is in a status of BROKEN, it is not possible to start the desired subsystem until this problem is fixed.
- **Automation flags are turned off**

Automation flags may be turned off for the subsystem or for just the INITSTART or RESTART actions. If the INITSTART flag is turned off, the subsystem cannot be started from a DOWN status. If the RESTART flag is turned off, the subsystem cannot be started from a RESTART status.

To see the flag settings for a subsystem you can use either DISPFLGS or DISPINFO. DISPFLGS will show you both the current settings in force for the subsystem and, perhaps more importantly, the settings that were made that resulted in those values.

- **Trigger is not satisfied**

  Triggers tell SA z/OS when it is allowed to stop and start subsystems. If a trigger has been linked to a subsystem, SA z/OS will not stop or start that subsystem unless the trigger says it is OK to do so. Normally this function is fine, but there may be times when you need to override it.

  To find out about the trigger linked to a subsystem, use the DISPTRG command. For example, the following command shows the triggers and events that are associated with the resource TEST0:

  ```
  DISPTRG TEST0,TARGET=KEY3
  ```

  If you want to start something and override the event settings you can either issue an INGREQ REQ=START OVERRIDE=TRG command or with the INGREQ command, you can specify that trigger conditions should not be considered by specifying option *Override ==> TRG* in the INGREQ command dialog panel. If you want to ensure that the trigger is satisfied before you start the resource, you can set it with INGEVENT. INGEVENT can be used to set an event (but may affect more than just the subsystem you are dealing with).

- **The automation manager automation flag is off:**

  Use the INGINFO command or the INGLIST command to find out what the current setting is. If the automation flag is off, the automation manager will not send the order to start the subsystem to the automation agent.

- **The subsystem is suspended:**

  Use the INGINFO command or the INGLIST command to find out what the current status is. If the subsystem is suspended, the automation manager will not send the order to start the subsystem to the automation agent.

  If you want to start a subsystem and override its suspend request, you can either issue an INGREQ REQ=START OVERRIDE=SUS command or specify option *Override ==> SUS* in the INGREQ command dialog panel to ignore the suspend requests.

If Resources Do Not Stop

If you have tried to stop a subsystem and SA z/OS does not do so, check for the following:

- **Use INGWHY command to initially analyze the situation:**

  Use INGWHY to see possible reasons for the current state of the resource. For more details, see *IBM Z System Automation Operator's Commands*.

- **Supporting resource cannot stop:**

  The supporting resources cannot be brought into the condition fulfilling the stop dependency, so the dependent resource that you want to stop also cannot stop. This is the case if the compound status indicates AWAITING.

- **Request with higher priority to have the subsystem UP:**

  Use the INGVOTE command to determine whether a higher priority request exists for the subsystem. If so, you can either reissue the stop request with a higher priority, thus overruling the other request, or cancel the other request.

- **The shutdown trigger is not satisfied:**

  Use the DISPTRG command to find out the trigger conditions that are defined for the subsystem. View the shutdown conditions that are defined for the subsystem. You can either set one or more events that

bring the shutdown condition into a satisfied state or use the INGREQ OVERRIDE=TRG parameter to bypass the trigger check.

- **Automation flags turned off:**

  The automation flag may be turned off for the subsystem or the termination action. To view the flag settings, use DISPSTAT, DISPFLGS, or DISPINFO.

- **The automation manager automation flag is off:**

  Use the INGINFO command or the INGLIST command to find out what the current setting of the automation manager automation flag is. If the automation flag is off, the automation manager will not send the order to stop the subsystem to the automation agent.

- **The subsystem is suspended:**

  Use the INGINFO command or the INGLIST command to find out what the current status is. If the subsystem is suspended, the automation manager will not send the order to stop the subsystem to the automation agent.

  If you want to stop a subsystem and override its suspend request, you can either issue an INGREQ REQ=STOP OVERRIDE=SUS command or specify option *Override ==> SUS* in the INGREQ command dialog panel to ignore the suspend requests.

- **Check history:**

  Use the INGINFO command to analyze the history of the resource.

Availability Scenario

About this task

Consider the following scenario:

You get a phone call from one of your users: Resource RVBASIC is not available. Find out why and start it.

Here are the steps to make the resource available:

Procedure

1. In this step, find out the details about RVBASIC resource, especially its status information.
   a) Display details of RVBASIC, using the **`INGLIST RVBASIC`** command.

   Suppose you get a following output screen. RVBASIC has a compound status of SATISFACTORY. It results from the fact that the observed status, SOFTDOWN, fulfills the desired status of UNAVAILABLE even though you expect it to be available. This panel also shows that RVBASIC is a BASIC APG, which can be available until all its remembers are available.

```
INGKYST0                 SA z/OS  - Command Dialogs     Line  1    of 1
Domain Id   = AOFDA      -------- INGLIST   ---------    Date = 01/13/19
Operator Id = SADEMO           Sysplex = SYSPLEX1         Time = 10:28:23
 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    Compound      Desired      Observed    Nature
 --- ------------ ---- --------  ------------  -----------  ----------  --------
 __  RVBASIC      APG  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN    BASIC
```

   b) Display the members of RVBASIC APG to investigate more details, by using the G (Members) action code. Enter g in the **CMD** column next to RVBASIC, and press enter.

Suppose you get an output screen as below. It shows that RVBASIC group has four members, RV01, RV02, RV05, and RV06, which are all SOFTDOWN. These members must be available, before the RVBASIC group can be available.

```
 INGKYST0                 SA z/OS  - Command Dialogs     Line  1    of 1
 Domain Id   = AOFDA      -------- INGLIST   ---------    Date = 01/13/19
 Operator Id = SADEMO          Sysplex = SYSPLEX1         Time = 10:28:57
  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    Compound      Desired      Observed    Nature
 --- ------------ ---- --------  ------------  -----------  ----------  --------
 __  RV01         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
 __  RV02         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
 __  RV05         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
 __  RV06         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
```

c) Display all the affected resources that you're going to start using the following command, as they all start with "RV".

```
INGLIST RV*
```

RVBASIC and its members are displayed.

```
 INGKYST0                 SA z/OS  - Command Dialogs     Line  1    of 1
 Domain Id   = AOFDA      -------- INGLIST   ---------    Date = 01/03/19
 Operator Id = SADEMO          Sysplex = SYSPLEX1         Time = 10:29:06
  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    Compound      Desired      Observed    Nature
 --- ------------ ---- --------  ------------  -----------  ----------  --------
 __  RVBASIC      APG  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN    BASIC
 __  RV01         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
 __  RV02         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
 __  RV05         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
 __  RV06         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
```

2. In this step, use the INGWHY command to find out what caused RVBASIC group to be unexpectedly unavailable (SOFTDOWN). Enter x in the **CMD** column next to it, and press Enter.

   The following panel is displayed, which shows that RVBASIC was stopped intentionally by the operator (SASTUD1). So to start RVBASIC, you need to remove this operator stop request.

```
INGKYST0                 SA z/OS  - Command Dialogs     Line  1    of 1
Domain Id   = AOFDA      -------- INGLIST   ---------    Date = 01/03/19
Operator Id = SADEMO          Sysplex = SYSPLEX1         Time = 10:29:15

Analyzed Resource: RVBASIC/APG/MVSA
Status   Compound: SATISFACTORY  Desired: UNAVAILABLE      Observed: SOFTDOWN
--------------------------------------------------------------------------------

 SITUATION:
RVBASIC/APG/MVSA is unavailable.

 REASON:
RVBASIC/APG/MVSA is desired to be UNAVAILABLE.
RVBASIC/APG/MVSA got a MAKEUNAVAILABLE request.
The request was issed by OPERATOR SASTUD1.

 ACTION:                                                 INGWHYSA(A0107401)
Refer to your company's rules in order to take the appropriate action.
No action required.
```

3. In this step, remove the stop request of RVBASIC to allow it to become available again.
   a) Press PF3 to go back to the INGLIST panel, and enter e (INGVOTE) in the **CMD** column next to RVBASIC to display its vote.

      From the following displayed INGVOTE panel, you can see that there is an operator MakeUnAvailable request to stop the resource, which has obviously been satisfied.

```
INGKYRQ0                  SA z/OS  - Command Dialogs      Line  1    of 5
Domain Id   = AOFDA     ---------- INGVOTE  ----------    Date = 01/03/19
Operator Id = SADEMO          Sysplex = SYSPLEX1          Time = 10:29:51

 Resource   ==>  RVBASIC/APG/MVSA____________
 System     ==>  ________  System name, Domain Id or sysplex name

 Desired Availaible..: Always

Cmd:  C cancel request   K  Kill request    S show request details
Cmd Action WIN Request/Vote Data
--- ------ --- ------------------------------------------------------------
__   STOP   Y  Request    : MakeUnAvailable
               Created    : 2019-01-02  22:43:25
               Originator : OPERATOR(SASTUD1)
               Priority   : 01720000     Should Be Down - Operator
               Status     : Winning/
Satisfied
```

   b) Remove the stop request either using option C (cancel request), which prompts you to verify the affected resources before canceling the request, or using option K (kill request) without verification.

      The following panel shows that the stop request is removed.

```
INGKYRQ0                  SA z/OS  - Command Dialogs      Line  1    of 5
Domain Id   = AOFDA     ---------- INGVOTE  ----------    Date = 01/03/19
Operator Id = SADEMO          Sysplex = SYSPLEX1          Time = 10:30:04

 Resource   ==>  RVBASIC/APG/MVSA____________
 System     ==>  ________  System name, Domain Id or sysplex name

 Desired Availaible..: Always

Cmd:  C cancel request   K  Kill request    S show request details
Cmd Action WIN Request/Vote Data
--- ------ --- ------------------------------------------------------------
   **CANCELLED**
               Created    : 2019-01-10  22:43:25
               Originator : OPERATOR(SASTUD1)
               Priority   : 01720000     Should Be Down - Operator
               Status     : Winning/Satisfied
```

   c) Press PF3 to return to INGLIST panel, and press PF9 to refresh the panel.

      Suppose the following panel is displayed. You can see that only the RV02 member is started, and the compound status of RVBASIC group is INAUTO, not expected SATISFACTORY status. Why INAUTO? This needs further investigation.

```
 INGKYST0                  SA z/OS  - Command Dialogs     Line  1    of 1
 Domain Id   = AOFDA      -------- INGLIST   ---------    Date = 01/03/19
 Operator Id = SADEMO          Sysplex = SYSPLEX1        Time = 10:30:16
  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    Compound      Desired      Observed    Nature
 --- ------------ ---- --------  ------------  -----------  ----------  --------
 __  RVBASIC      APG  MVSA      INAUTO        AVAILABLE    STARTING    BASIC
 __  RV01         APL  MVSA      AWITING       AVAILABLE    SOFTDOWN
 __  RV02         APL  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE
 __  RV05         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
 __  RV06         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
```

4. In this step, use INGWHY command to further investigate why RVBASIC cannot start after its stop vote is removed.

   a) Enter x next to RVBASIC and press Enter. The first panel shows that RVBASIC is desired to be always available, but tells no clue why it cannot be available.

```
 INGKYAN2                  SA z/OS  - Command Dialogs     Line  1     of 17
 Domain Id . : AOFDA   ------------ INGWHY   ------------ Date . . : 01/03/19
 Operator Id : SADEMO             Sysplex = SYSPLEX1      Time . . :
10:30:27

 Analyzed Resource: RVBASIC/APL/MVSA
 Status   Compound: INAUTO        Desired: AVAILABLE      Observed: STARTING
 ------------------------------------------------------------------------------
  SITUATION:
 RVBASIC/APG/MVSA is starting or waiting to be
started.

  REASON 1 of 4:
 The 'Desired Availability' of RVBASIC/APG/MVSA is set to 'Always'.
 RVBASIC/APG/MVSA is started by default once its dependencies are satisfied.

  ACTION:                                                  INGWHYSA(A0106900)
 Refer to your company's rules in order to take the appropriate action.
 No action required.
```

   b) Press PF11 to display Reason 2. As the following panel shows, the member RV05 was stopped by the RV56SVP schedule, which prevents RVBASIC from starting.

```
 SITUATION:
RVBASIC/APG/MVSA is starting or waiting to be started.

 REASON 2 of 4:
RVBASIC/APG/MVSA has a dependency on RV05/APL/MVSA.
RV05/APL/MVSA is desired to be UNAVAILABLE becasue of a MAKEAVAILABLE vote.
The vote was caused by SCHEDULE RV56SVP.

ACTION:INGWHYSA(A0107200)
Refer to your company's rules in order to take the appropriate action.
No action required.
```

   c) Press PF11 to display Reason 3. It shows that the member RV01 cannot start because a startup event is unset for its trigger RV01TRIG.

```
 SITUATION:
RVBASIC/APG/MVSA is starting or waiting to be started.

 REASON 3 of 4:
RVBASIC/APG/MVSA has a dependency on RV01/APL/MVSA.
RV01/APL/MVSA is desired to be Available.
RV01/APL/MVSA is controlled by trigger RV01TRIG.
A startup event of RV01TRIG is unset.

 ACTION:                                                  INGWHYSA(A0702800)
Refer to your company's rules in order to take the appropriate action.
Consider the following commands to investigate RV01/APL/MVSA:
- DISPTRG
- INGEVENT
- INGTRIG
```

d) Press PF11 again to display Reason 4. As same as RV05 in Reason 2, RV06 is also stopped by the same schedule RV56SVP.

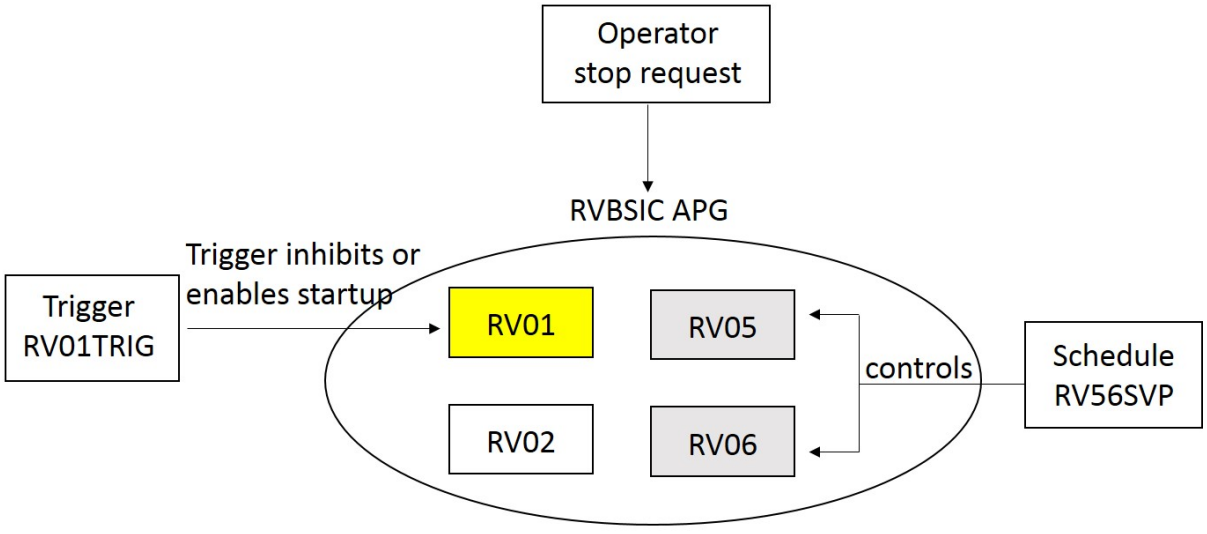
```
 SITUATION:
RVBASIC/APG/MVSA is starting or waiting to be
started.

 REASON 4 of 4:
RVBASIC/APG/MVSA has a dependency on RV06/APL/MVSA.
RV05/APL/MVSA is desired to be UNAVAILABLE becasue of a MAKEAVAILABLE vote.
The vote was caused by SCHEDULE RV56SVP.

 ACTION:                                                  INGWHYSA(A0107200)
Refer to your company's rules in order to take the appropriate action.
No action required.
```

In summary, INGWHY shows the following reasons that inhibits RVBASIC from starting.

- RV01 has a trigger connected to it. One startup condition for that trigger must be fulfilled to make RV01 available. See step “5” on page 205.
- RV05 and RV06 are both controlled by a schedule. The downtime schedule need to be modified. See step “6” on page 206.



5. In this step, you start RV01, whose trigger is not satisfied.
   a) Display the trigger of RV01 using the DISPTRG command. Enter action code h against RV01 and press Enter. It displays the triggers and their events for the selected resource.

In the following displayed panel in this scenario, you can see that trigger RV01EVT has a startup condition that is fulfilled if event RV01EVT is set. This is what you need to find out in the next step.

```
AOFLT000                 SA z/OS  - Command Dialogs      Line  1    of 1
Domain Id   = AOFDA     ---------- DISPTRG  ----------    Date = 01/03/19
Operator Id = SADEMO          System  = MVSA              Time = 10:31:58

 Resource   ==>  RV01/APL/MVSA_____________
 Target     ==>  __________      System name, Domain Id or sysplex name
 Trigger. . . :  RV01TRIG                      Observed status :  SOFTDOWN

Cmd:  S show details
Cmd Type      Events
--- --------  -------- -------- -------- -------- -------- -------- ---------
 s  STARTUP   RV01EVT
```

b) To see more details about an event (or the condition), enter action code s in the **CMD** column.

The following panel is displayed. You can see that the event RV01EVT is not set (the **Status** column contains UNSET).

```
AOFLT100                 SA z/OS  - Command Dialogs      Line  1    of 1
Domain Id   = AOFDA    --- Trigger Condition List ----    Date = 01/03/19
Operator Id = SADEMO          System  = MVSA              Time = 10:32:02

 Trigger . . . . : RV01TRIG
 Condition type. : STARTUP
 Resource. . . . : RV01/APL/MVSA             Observed status : SOFTDOWN

Cmd:  D show resources   S set event for resource   U unset event for resource
Cmd Event     Status       Unset  Description
--- --------  -----------  -----  ---------------------------------------------
 s_ RV01EVT   UNSET        UP
```

c) Enter s next to the event and press Enter. This step sets the event to fulfill the startup condition for trigger RV01TRIG, thus allowing the resource RV01 to start.

d) Press PF3 multiple times until you return to the INGLIST panel, and press PF9 to refresh. You can see that RV01 is now available.

```
CMD Name         Type System    Compound      Desired      Observed    Nature
--- ------------ ---- --------  ------------  -----------  ----------  --------
__  RVBASIC      APG  MVSA      INAUTO        AVAILABLE    STARTING    BASIC
__  RV01         APL  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE
__  RV02         APL  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE
__  RV05         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
__  RV06         APL  MVSA      SATISFACTORY  UNAVAILABLE  SOFTDOWN
```

6. In this step, you start RV02 and RV05, which are both stopped by the RV56SVP schedule.

a) Display the schedules that are defined in this sysplex by entering the INGSCHED command. RV56SVP schedule is displayed.

```
INGKYSP0                 SA z/OS  - Command Dialogs       Line  1    of 1

Domain Id . : AOFDA   ------------ INGSCHED ------------   Date . . : 01/03/19
Operator Id : SADEMO             Sysplex = SYSPLEX1        Time . . : 10:32:47

Cmd:  A Show Details   B Show Overrides   C Show Resources

Cmd Schedule  Description
--- --------  ---------------------------------------
__  RV56SVP   Schedule for RV05 and RV06
```

b) Enter C against the schedule to show the resources linked to it. You can see that only RV05 and RV06 use the schedule.

```
INGKYSP3                SA z/OS - Command Dialogs        Line 1     of 2
Domain Id . : AOFDA   ------------ INGSCHED ------------   Date . . : 01/11/19
Operator Id : SADEMO             Sysplex = SYSPLEX1        Time . . : 10:32:55

Schedule. . : RV56SVP      Schedule for RV05 and RV06

Cmd:  S Show Overrides
Cmd Resource     Type  System    Description
--- --------     ----  --------  ---------------------------------------
__  RV05         APL   MVSA      RV05 Appl has service period RV56SVP
__  RV06         APL   MVSA      RV06 Appl has service period RV56SVP
```

c) Use the S action code to check if any resource override respectively for both RV05 and RV06.

Suppose the following service windows are displayed for both resources, and the service windows are in green color. Green color means that there are no overrides to the base schedule. You can see that the current time (10:33:01) is within today's requested downtime (0100 - 2100). Then, you need to override the RV56SVP base schedule to start RV05 and RV06.

```
INGKYSP2                SA z/OS  - Command Dialogs
Domain Id   = AOFDA     ---------- INGSCHED ----------    Date = 01/03/19
Operator Id = SADEMO          Sysplex = SYSPLEX1          Time = 10:33:01

Resource . . . : RV05/APL/MVSA               Schedule . . :  RV56SVP
Starting date =>  01 / 03 / 2019     (mm/dd/yyyy)

C Date            Pri  From-To    From-To    From-To    From-To    From-To
- ------------ -- ---  ---------  ---------  ---------  ---------  ---------
  01/03/19 Tue UP  L_  ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_  0100 2100  ____ ____  ____ ____ ____ ____ ____ ____
  01/04/19 Wed UP  L_  ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_  0100 2100  ____ ____  ____ ____ ____ ____ ____ ____
  01/05/19 Thu UP  L_  ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_  0100 2100  ____ ____  ____ ____ ____ ____ ____ ____
  01/06/19 Fri UP  L_  ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_  0100 2100  ____ ____  ____ ____ ____ ____ ____ ____
```

d) Press PF3 until you return to the INGSCHED panel, enter b against RV56SVP schedule, and press Enter to show overrides of the schedule.

```
INGKYSP0                SA z/OS  - Command Dialogs       Line  1    of 1
Domain Id . : AOFDA   ------------ INGSCHED ------------   Date . . : 01/03/19
Operator Id : SADEMO             Sysplex = SYSPLEX1        Time . . : 10:33:17

Cmd:  A Show Details   B Show Overrides   C Show Resources

Cmd Schedule  Description
--- --------  ---------------------------------------
b_  RV56SVP   Schedule for RV05 and RV06
```

e) Overwrite the end time of today's downtime to some minutes later, and press Enter. A FUNCTION COMPLETED message is displayed to show the override success. In this scenario, the current time is 10:33, and you can override the end time to 10:34. When it turns 10:34, SA will start RV05 and RV06, as both resources get a default MakeAvailabe vote from the RVBASIC group.

```
INGKYSP2                SA z/OS  - Command Dialogs
Domain Id   = AOFDA     ---------- INGSCHED ----------    Date = 01/03/19
Operator Id = SADEMO          Sysplex = SYSPLEX1          Time = 10:33:20

Resource . . . : RV05/APL/MVSA                Schedule . . :  RV56SVP
Starting date =>  01 / 03 / 2019     (mm/dd/yyyy)

C Date            Pri  From-To    From-To    From-To    From-To    From-To
- ------------ -- ---  ---------  ---------  ---------  ---------  ---------
  01/03/19 Thu UP  L_   ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_   0100 1034  ____ ____  ____ ____ ____ ____ ____ ____
  01/04/19 Fri UP  L_   ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_   0100 2100  ____ ____  ____ ____ ____ ____ ____ ____
  01/05/19 Sat UP  L_   ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_   0100 2100  ____ ____  ____ ____ ____ ____ ____ ____
  01/06/19 Sun UP  L_   ____ ____  ____ ____  ____ ____ ____ ____ ____ ____
               DN  L_   0100 2100  ____ ____  ____ ____ ____ ____ ____ ____
```

f) Press PF3 until you return to the INGLIST panel, wait until the end time you specified (10:34 in this case), and then press PF9 to refresh the panel.

You can see RV02 and RV05 are now started.

```
INGKYST0                 SA z/OS  - Command Dialogs     Line  1    of 1
Domain Id   = AOFDA      -------- INGLIST   ---------    Date = 01/03/19
Operator Id = SADEMO          Sysplex = SYSPLEX1         Time = 10:34:02
 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    Compound      Desired      Observed    Nature
--- ------------ ---- --------  ------------  -----------  ----------  --------
__  RVBASIC      APG  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE   BASIC
__  RV01         APL  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE
__  RV02         APL  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE
__  RV05         APL  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE
__  RV06         APL  MVSA      SATISFACTORY  AVAILABLE    AVAILABLE
```

Results

RVBASIC is now available, after all its members are available.

Appendix A. SA-BCPii Use Cases

This appendix provides common use cases that illustrate how IBM Z System Automation customization settings can help to use the connection protocol more efficiently. IBM Geographically Dispersed Parallel Sysplex (GDPS) related information is included.

Base Control Program internal interface (BCPii) functional characteristics

BCPii is a communication protocol between an application running in a Logical Partition (LPAR) of an IBM Z mainframe (processor) under control of z/OS and the processor attached Support Element (SE). The proprietary protocol does not depend on your network infrastructure and does not use the IBM Z Channel Subsystem. Applications can issue operations management queries and hardware commands to monitor and control the processor and its facilities. The SE either returns immediate command responses or forwards hardware events to registered applications. Local and remote SEs connected to the same processor network can be targeted. In this case, the local SE acts as a gateway for the BCPii request and a Hardware Management Console (HMC) is used to forward the request to the remote SE. The routing HMC also sends data to the local SE of applications previously registered to receive event information.

Available BCPii implementations and general considerations

BCPii is a z/OS provided feature (HWI-BCPii) that offers its APIs to any supported z/OS application environment.

For the SA-BCPii implementation, requests must be issued by an application running under z/OS NetView control and is using System Automation internal services like GDPS, or APIs like ProcOps. Both BCPii implementations use the same services to communicate with the SE. For the IBM Z firmware, the BCPii implementation is transparent. The information source about the processor and its facilities for the SA-BCPii is the Simple Network Management Protocol (SNMP) database, provided by the SE console application. In an IBM Z processor network and on individual processors level, the remote and local BCPii related session workload might vary. It depends on the number of active BCPii applications and the amount of data that flows concurrently. In the BCPii request processing path, there is serialization and queuing in place, where needed. Considering BCPii performance, it is a good practice to keep track of the permanent BCPii sessions and the event traffic they produce.

System Automation functions and solution offerings using SA-BCPii connections

For information about how to prepare and configure the hardware for SA-BCPii connections, see the *Planning and Installation* manual.

For information about how to specify the connection credentials, processor and LPAR associations, and session monitoring information for SA-BCPii in the System Automation policy database, see the *Defining Automation Policies* manual.

Note: For LPAR Management, a ProcOps command emulation for a hardware command sub-set is no longer strategic and will not be further discussed. With SA z/OS 4.2, BCPii has been fully integrated into ProcOps. You are now encouraged to use native ProcOps commands to manage LPARs over SA-BCPii.

SA-BCPii provides different connection types:

INTERNAL
: It is required by GDPS and specific SA sysplex automation functions. From any defined z/OS system in the SA policy, connections to any defined IBM Z processor can be established, if the security settings and hardware related permissions allow to. Because GDPS and SA use the same SA-BCPii session, special serialization is needed and in place.

Hybrid SNMP
: This is a ProcOps specific connection alternative to SNMP over TCP/IP. ProcOps redirects operations management requests to SA-BCPii, instead of using TCP/IP. The SA ProcOps focal point function must be active before hybrid SNMP redirection is possible. From the SA ProcOps focal point system, a

BCPii connection to any other defined IBM Z processor can be established, if the security settings and hardware related permissions allow to. Hybrid SNMP sessions (ISQET32) are separate SA-BCPii connections. They operate in parallel to existing INTERNAL sessions and might target the same processor.

INTERNAL session control transfer between GDPS and System Automation

System Automation automatically starts all defined INTERNAL connections at SA agent start time. If processor related information in a policy data base (PDB) was changed, a refresh with the new build and loaded data at SA runtime might also cause INTERNAL connections to be restarted.

With GDPS 4.2, the solution offerings using the SA-BCPii no longer depend on System Automation to perform the session initialization for processors that are in the management scope of GDPS. Instead, System Automation detects and validates the SA-BCPii connection to the local SE and signals GDPS whether SA-BCPii and the local SE connection can be used. For an SA agent instance, this first step is critical to the overall availability of SA-BCPii. Only specific SA-tasks are authorized to use the SA-BCPii during that time. After SA has notified GDPS about the SA-BCPii status, GDPS can request which of the SA policy defined processor connections it wants to manage and monitor. Later, if GDPS wants to return session management responsibility back to SA, it can issue a message to request this.

The following table lists the messages exchanged between System Automation and GDPS to transfer INTERNAL connection control:

Table 18. Messages exchanged between SA and GDPS

Message IDs	Description
ING825I	Informs that GDPS can now request control of INTERNAL connections.
ING826I	Confirms that GDPS has control of a specific INTERNAL connection.
ING827I	Confirms that System Automation has control of a specific INTERNAL connection.
ING828I	Informs GDPS that the local SE connection cannot be used and therefore SA-BCPii is not available.
GEO2735I	Requests control for the specified INTERNAL connection, together with the overall number of the intended connection control requests.
GEO2736I	Returns control of a specific INTERNAL connection back to System Automation.

For more details about the messages, refer to the *GDPS and System Automation Messages and Codes* manuals.

Use Case: GDPS is managing your mainframes

You can specify policy settings to reduce the INTERNAL connection related workload when GDPS is managing your mainframe environment.

1. To reduce the INTERNAL connection related workload in IBM Z System Automation, specify the following PROCESSOR INFO policy settings in your PDB:

```
Connection Protocol . . . . . INTERNAL__  (INTERNAL SNMP)
Connection Monitor Interval . . NONE_     (mm:ss NONE)
```

   The session is managed and monitored by GDPS, and thus no connection monitoring in SA z/OS is needed.

2. Specify the processor's entry name (T35 in this example) as the only connection token so SA z/OS will not try to initialize connections to other processors if GDPS fails or is delayed in sending GE02735I request messages to SA z/OS.

```
Connection Scope . . . . . . . T35_____
```

3. In your **CNMSTYLE** definitions, ensure that **COMMON.AOF_AAO_HW_VALIDATION** is set to NO. By default, the hardware validation function which is bound to the INTERNAL protocol is not enabled. If you previously set it to YES, change it to NO.

```
COMMON.AOF_AAO_HW_VALIDATION = NO
```

Use Case: ProcOps over SA-BCPii is managing your mainframes

In contrast to INTERNAL connections, a ProcOps hybrid SNMP session to a processor has all important hardware event notifications permanently enabled.

Therefore, specify moderate polling frequencies can avoid losing status information.

```
Connection Protocol . . . . . . SNMP__   (INTERNAL SNMP)
Path Poll Frequency . . . . . . 30__   (0 to 99 minutes)
```

ProcOps acts as a Focal Point (FP) targeting one or multiple processors, thus the overall number of hybrid SNMP connections used is less compared to the total number of INTERNAL connections in an any-to-any SA-BCPii network.

Use Case: Options to limit the number of SA-BCPii connections

By default, IBM Z System Automation tries to connect to all INTERNAL processors defined in your PDB, unless these processors are managed by GDPS. It sets up an any-to-any SA-BCPii network between all participating SA agents sharing a PDB. There might be technical, site, or organization restrictions, especially if you use a common PDB to maintain all processors definitions in your enterprise, you might need a better way to control the automatic INTERNAL connection starts.

Connection Scope (INTERNAL only)
It specifies for each processor that the connections to what other defined processors should be established automatically.

```
Connection Scope . . . . . . . _____________
```

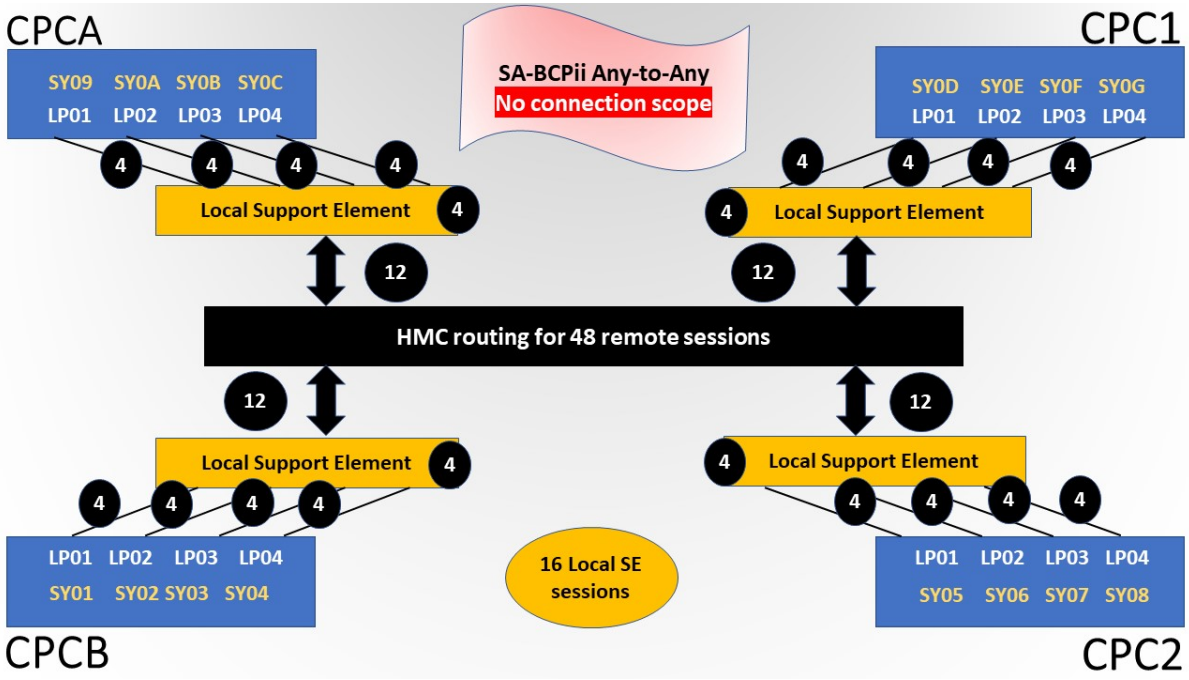
How does it work?

At SA z/OS start time or policy refresh time, if processor definitions have been changed, the connection scope definitions of all defined processors are read.

- Only if the local processor has a scope definition (one or multiple blank separated characters or token strings), a compare with the other scope definitions is made.
- If any of the local tokens are found in another processor's scope definition, a connection to this processor will be tried.
- If the local processor has no tokens defined, connections to all other processors will be tried, regardless of their scope definitions.
- If the local processor has token defined and no other processor shares the token, just the local processor connection will be started.

Sample case with no connection scope defined

Initially, the connection scope field in the policy for a processor is empty. SA agents running in LPARs of this processor will try to connect to all other processors defined with a connection protocol of INTERNAL. See the following sample case with no connection scope defined:



The numbers in the circles represent the sessions started by each SA agent and then split into local or remote connections.

Four processors CPC1, CPC2, CPCA, and CPCB are defined and run in the same processor network. Each one has a set of four LPARs with z/OS target systems assigned. All these systems are defined in the same policy database. When SA is up and running for all 16 systems, in total 64 SA-BCPii sessions will be active. On each processor, four sessions connect the local systems/LPARs. To control the remaining three processors, each system needs three more remote sessions, which use an HMC for routing. In total, there are 48 remote sessions.

It is better to split the bulk of SA-BCPii sessions into smaller segments to allow only systems in the same Sysplex to connect automatically. In the following example, connection scope definitions are used to achieve this.

Sample case with connection scope defined

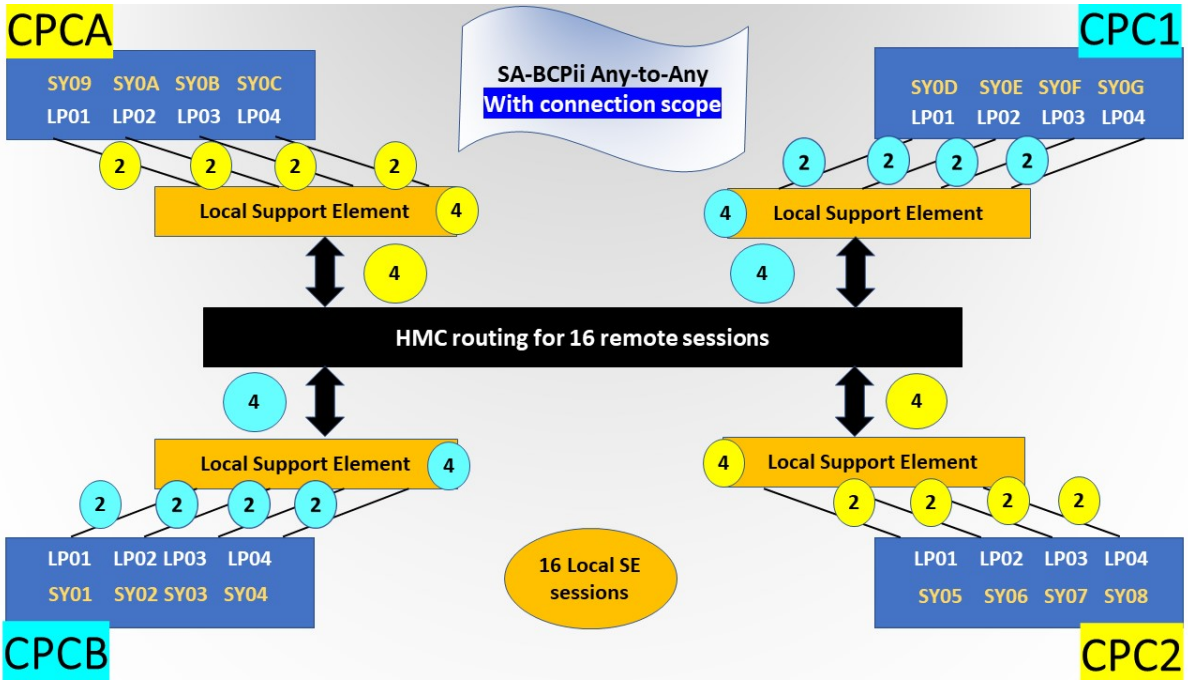
Processors CPCA and CPC2 share the following scope definition:

```
Connection Scope . . . . . PLEXA2
```

Processors CPCB and CPC1 share the following scope definition:

```
Connection Scope . . . . . PLEXB1
```

With connection scope defined, the number of INTERNAL sessions is reduced in the following sample case:



The numbers in the circles represent the sessions started by each SA agent and then split into local or remote connections.

Since the processor definitions have connection tokens defined, SA will only start an INTERNAL session automatically, if the determined token is also found in another processor's policy connection scope definition. The session to the own processor is implicitly started. With the connection tokens defined, systems on CPCB do not connect to CPC2 and CPCA. Likewise, systems on CPCA do not connect automatically to CPCB and CPC1. Despite connection scope definitions, you can start INTERNAL connections manually using command **`INGHWSRV processor CONNECT`**, where *processor* is the name of a processor defined in the PDB. If the connection started manually is not in the defined scope of the processor/system, where the **`INGHWSRV CONNECT`** request is issued, informational message `ING824I` is issued and the request continues.

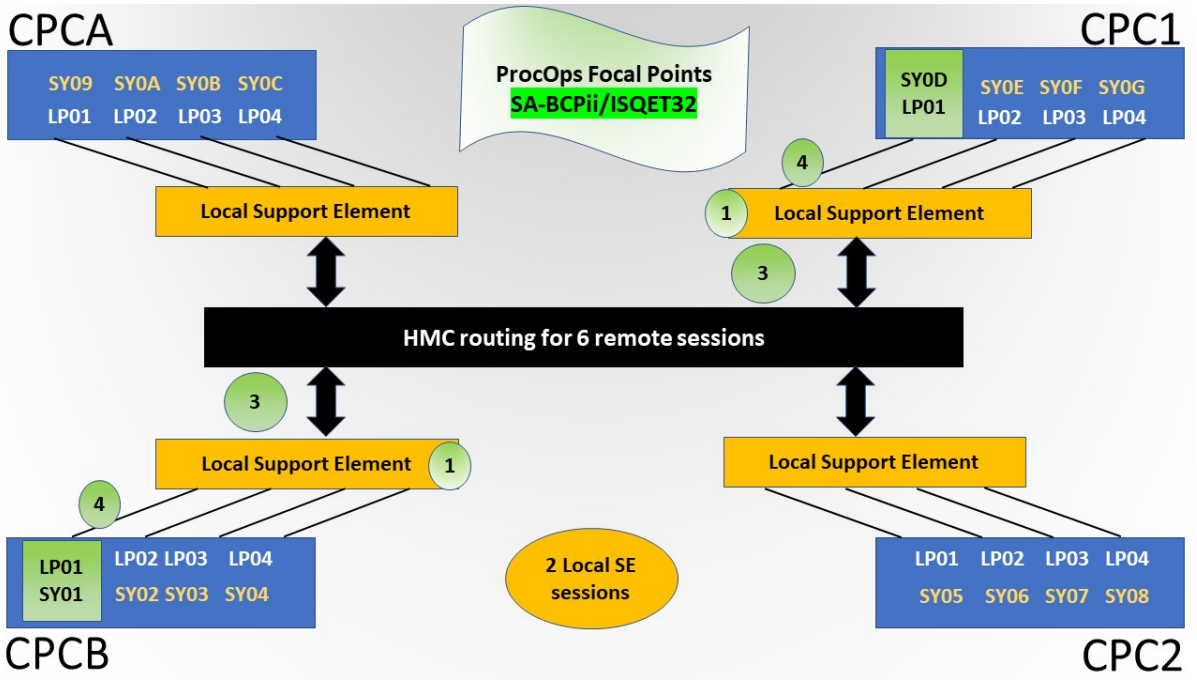
Your processor connection scope definition can have multiple tokens specified, for example:

```
The following specifications are for INTERNAL processors only:
  Auth Token  . . . . . . . . . . MSYSXXXX
  Connection Monitor Interval . . NONE       (mm:ss NONE)
  Connection Scope  . . . . . . . M1 X DBSYS
```

The SA agents running in an LPAR of a processor with these connection scope tokens will automatically try to connect to other processors defined in the SA policy that at least have one matching token (M1, X, or DBSYS). This way, an SA agent can have connections to processors with different connection scopes.

Connection focal points (ProcOps hybrid SNMP only)

The number of required SA-BCPii connections can be further reduced by using a ProcOps focal point (FP) with hybrid SNMP. See the following sample case:



The numbers in the circles represent the sessions started by each ProcOps FP and then split into local or remote connections.

In the PROCESSOR INFO policy, processors, CPCA, CPCB, CPC1, and CPC2 have the following specifications:

```
Connection Protocol . . . . . SNMP          (INTERNAL SNMP)
 TCP/IP Address/Hostname or identifier ISQET32 for BCPii redirection
ISQET32
```

In the LPARS AND SYSTEMS policy of each processor, the LPARs with the associated system names are defined. Take processor CPCB as an example:

```
LPAR Name   Target Mode    Target System Name
LP01        ESA           SY01
LP02        ESA           SY02
LP03        ESA           SY03
LP04        ESA           SY04
```

In addition, in the SYSTEM policy for each target system, the PROCESSOR OPERATIONS SPECIFIC POLICY item TARGET SYSTEM INFO needs to be defined. Finally, in the PDB APL policies for SY01 and SY0D, you can define ProcOps as an application to automate start and stop of ProcOps and monitor its availability. Depending on your mode of operation, ProcOps FPs and their ISQET32 connections can be activated and deactivated on demand.

Use Case: Working with GDPS and ProcOps

ProcOps and GDPS use different SA-BCPii connection types to manage and monitor IBM Z mainframes.

Depending on the hardware and system configuration, it might be helpful to use ProcOps and GDPS together. Especially, if the available function sets do complement. When you plan to use both, consider the following items:

- Automation

  Do not implement contradicting automation by using both components at the same time to automate the same target resources. If you have issues like availability, resiliency, or recovery time, use only one as the primary hardware automation component. You can also split the control of your target resources between GDPS and ProcOps to avoid automation conflicts.

- Monitoring

  GDPS monitoring is based on permanent connection polling and temporary hardware action-based event handling. ProcOps monitoring is based on permanent connection polling and instant event detection for most of the events emitted by the IBM Z hardware, including hardware messages.
- Hardware Management Console (HMC) tasks

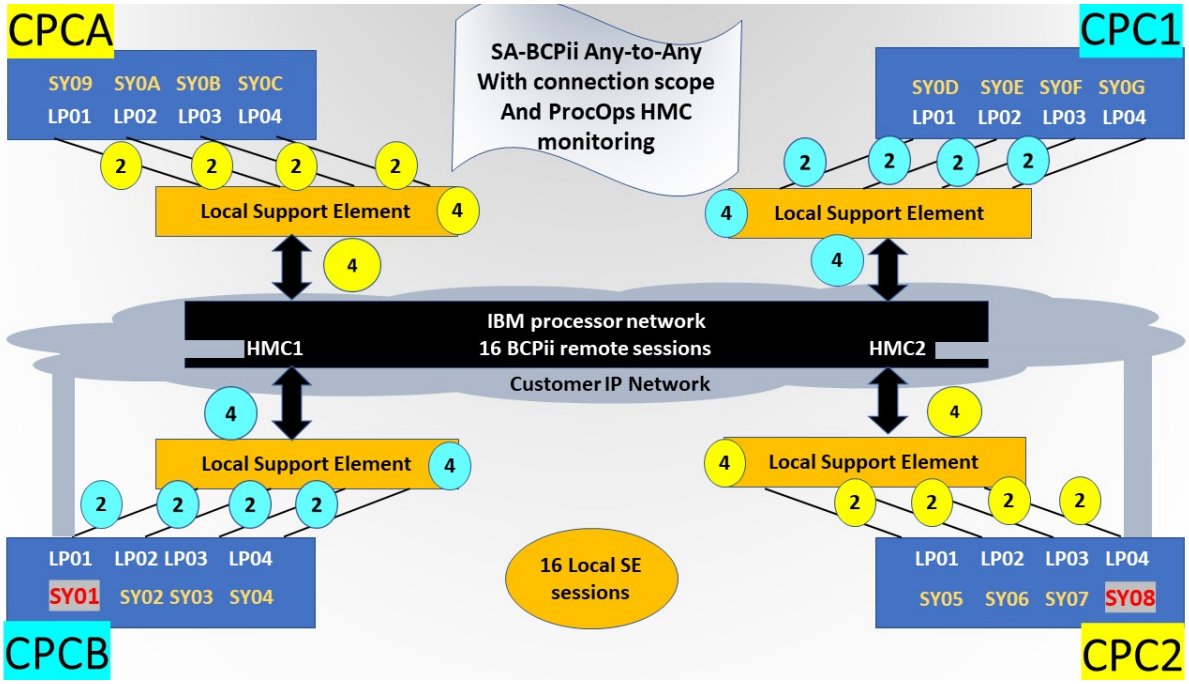
  Many of the HMC operators' or system programmers' tasks can be automated by using the ProcOps common command API **ISQCCMD**. This includes tasks that cannot be mapped in GDPS scripts or out of its mission.
- Monitor HMC availability for BCPii routing

  If the HMC is configured, ProcOps SNMP IP HMC connections can be used to monitor its BCPii routing capability.

Use Case: Monitoring the Hardware Management Consoles

Hardware Management Consoles (HMCs), which are configured to route BCPii requests, responses, and events, cannot be determined and monitored using BCPii.

See the following figure that illustrates how ProcOps SNMP over TCP/IP can be used:



In this SA-BCPii connection scope environment, HMC1 and HMC2 are configured for BCPii routing. For details, see section *Preparing the HMC* in the *Planning and Installation Guide*. The systems SYS01 and SY08 are configured as ProcOps focal points (FPs) and run in parallel to the SA-BCPii connections. Each HMC has a network adapter connected to the customer IP network. The two ProcOps FPs have sessions to both HMCs. In the PROCESSOR INFO policy, processors, CPCA, CPCB, CPC1, and CPC2 have the following specifications:

- CPC1, CPC2, CPCA, CPCB

```
Connection Protocol . . . . . SNMP INTERNAL   (INTERNAL SNMP)
```

- CPC1

```
The following specifications are for INTERNAL processors only:
  Auth Token  . . . . . . . . . . XXXXXXXX
  Connection Monitor Interval . . NONE      (mm:ss NONE)
  Connection Scope  . . . . . . . CPC1 CPCB
```

- CPC2

```
The following specifications are for INTERNAL processors only:
  Auth Token  . . . . . . . . . . . XXXXXXXX
  Connection Monitor Interval . . NONE      (mm:ss NONE)
  Connection Scope  . . . . . . . CPC2 CPCA
At least one address must be specified:
  TCP/IP Address/Hostname or identifier ISQET32 for BCPii redirection
_____________________________________________________________________
  Alternate Adress/Hostname or identifier ISQET32 for BCPii redirection
CPCHMC2_____________________________________________________________
```

- CPCA

```
The following specifications are for INTERNAL processors only:
  Auth Token  . . . . . . . . . . . XXXXXXXX
  Connection Monitor Interval . . NONE      (mm:ss NONE)
  Connection Scope  . . . . . . . CPC2 CPCA
```

- CPCB

```
The following specifications are for INTERNAL processors only:
  Auth Token  . . . . . . . . . . . XXXXXXXX
  Connection Monitor Interval . . NONE      (mm:ss NONE)
  Connection Scope  . . . . . . . CPC1 CPCB
At least one address must be specified:
  TCP/IP Address/Hostname or identifier ISQET32 for BCPii redirection
_____________________________________________________________________
  Alternate Adress/Hostname or identifier ISQET32 for BCPii redirection
CPCHMC1_____________________________________________________________
```

On the HMCs, the CPC1, CPC2, CPCA, and CPCB processors are members of the **Defined CPC Group**, including all LPARs of interest, which must be members of the **Defined CPC Image Group** of the HMCs. The HMC definitions must match the PDB definitions for processors and their LPARs and systems definitions.

Since both HMCs are enabled for BCPii routing and have the same set of CPCs and LPARs defined, a functional BCPii routing redundancy is available. If SA-BCPii sessions return timeout or heartbeat errors, check if the ProcOps SNMP TCP/IP connections to both HMCs are still working. Issue command **ISQXDST** or **ISQXCON** to display status summary. If you see SNMP SESSION PROBLEMS for processor connections on both HMCs and you cannot successfully initialize any session to either HMC, SA-BCPii routing is most likely not working because of the HMC failures.

Appendix B. Automation Initialization Options

At the start of SA z/OS initialization a multiline write-to-operator (MLWTO) message, AOF767I, is issued describing the available options. The lines of the MLWTO define the options that can be specified for SA z/OS initialization. Each line is listed and explained below.

AOF767I is followed by either an AOF603D or an AOF606D WTOR. SA z/OS replies automatically to an AOF603D after two minutes unless this has been changed as part of your site's automation policy. SA z/OS does not automatically reply to an AOF606D message.

The appearance of AOF767I, AOF603D and AOF606D can be suppressed by setting the variable AOFINITREPLY to 0 (see the table "Global Variables to Enable Advanced Automation (CGLOBALS)" in *IBM Z System Automation Customizing and Programming*).

A description of the options follows:

. STOP - CANCEL AUTOMATION
: This line is always present. It indicates that you can reply **STOP**. Doing so causes SA z/OS to issue an AOF605I and stop running. You must recycle NetView to restart SA z/OS.

. PAUSE - SUSPEND AUTOMATION
: This line is present unless SA z/OS initialization has previously been paused. It indicates that you can reply **PAUSE**. Doing so causes SA z/OS to go into a paused state. An AOF604I message is issued, the AOF767I message is reissued, and an AOF606D WTOR is issued. You should not automate the reply to the AOF606D WTOR, as this disables the pause function.

  If the automation manager is started, restarted, or refreshed (with the INGAMS or MVS modify command) while the automation agent is paused, the reply is cancelled and the automation agent initialization is redriven from the beginning.

. NOSTART - DO NOT AUTOMATE SUBSYSTEM STARTUP
: This line is always present. It indicates that you can reply **NOSTART**. Doing so causes SA z/OS to suspend the automation agent (see INGAMS) to avoid that any start or stop orders are sent by the automation manager.

. RUNMODE=x - SET RUNMODE(CURRENTLY *mode*)
: This line is present when runmodes are used in the policy or when a runmode request is present in the automation manager. It indicates that can reply RUNMODE= followed by a valid runmode. Doing so causes SA z/OS to set the specified runmode before the automation agent is resumed after loading the configuration. You can specify RUNMODE=? to get a list of all valid runmodes for the current system.

. ENTER - CONTINUE
: The initialization of SA z/OS continues.

A sample of how the AOF767I message could look, is in Figure 54 on page 217.

```
AOF767I AUTOMATION OPTIONS:
.STOP      - CANCEL AUTOMATION
.PAUSE     - SUSPEND AUTOMATION
.NOSTART   - DO NOT AUTOMATE SUBSYSTEM STARTUP
.RUNMODE=x - SET RUNMODE (CURRENTLY *ALL)
.ENTER     - CONTINUE
```

Figure 54. Sample AOF767I Message

If the message on your screen looks significantly different from the sample shown, ask your automation administrator whether they used the AOFEXI01 initialization exit to modify the message.

Appendix C. Problem Determination

This appendix contains various subtopics that deal with methods of avoiding or detecting problems during SA z/OS automation. The following topics are discussed:

Maintaining System Status during a Status Forwarding Path Failure

SA z/OS uses processor operations functions and XCF to attempt to maintain the current status of a system, even if there is a failure in the status forwarding path. SA z/OS system operations uses XCF for both passive and active monitoring of system status. SA z/OS is able to use XCF to maintain system status for all systems in the sysplex.

SA z/OS system operations receives alerts generated by processor operations for target systems monitored by processor operations. If, for example, a target system enters a wait state or a DCCF condition, SA z/OS will use the resulting alert from the processor operations functions to update the system status, even if the status forwarding path between SA z/OS and the target system is down.

Debugging and Tracing

This information describes the tracing and debugging facilities that are available with SA z/OS.

The following topics are discussed:

To collect debugging information you can also use the command INGLKUP REQ=COLLECT.

Automation Manager State Trace Table

The event handler trace back table is written to SYSLOG or to the Message Logger. It is the most important debug tool when you want to understand the event flow.

The table is written using the HSAM1399I message due to one of the following:

- Address space termination (written automatically to the Message Logger)
- `/F jobname,$TRACESTATE` is written to SYSLOG

Using Trace Services for the Automation Manager and Agent

To trace the control flow of a process, SA z/OS uses the **MVS Component Trace Facility** with all its capabilities.

This requires an External Writer as soon as the trace is started.

Before the trace can be activated, the trace data set must be allocated. The recommended data set name is `'SYS1.HSA.sysname.CTRACE01'`. An appropriate JCL is provided in the SINGSAMP SA z/OS sample library as the member HSACTWR. In addition, the JCL of the component trace's external writer must be

cataloged in SYS1.PROCLIB. For more information on customizing the component trace, refer to *IBM Z System Automation Planning and Installation*.

To collect the trace data in a data set, the external writer must be started. Enter the following command on a z/OS console:

```
TRACE CT,WTRSTART=HSACTWR
```

To activate the trace, enter the following command on a z/OS console:

```
TRACE CT,ON,COMP=HSAAM
```

When the system responds with a WTOR for the trace options, reply as follows:

```
xx,WTR=HSACTWR,OPTIONS=(ALL|set_of_options),END
```

Where *set_of_options* is one of the options in Table 19 on page 220.

Table 19. SA z/OS Trace Activation Options

Option	**Activates the Trace In:**
ALL	All of the subsequent options
B[ase]	SA z/OS service routines that use z/OS system services
COM[munication]	The communications framework
CON[trol]	The base framework
EN[gine]	Internal use
EV[ent]	The Sysplex Event Handler
F[ramework]	The service sustaining the LE environment
HA[rdware]	Processor operations processing
HI[story]	The component dealing with the history of work items
I[nstruction]	The services encoding or decoding the instruction streams
JLM	The job log monitoring task and related commands
L[ock]	The service creating and deleting locks
REQ[uest]	The beginning and ending of a work item (request) processing
REXX	REXX function routines
SCH[edule]	The service dealing with service periods
SDF	The SDF task and related commands
W[orkitem]	All components working on work items

Deactivate the trace with the command:

```
TRACE CT,OFF,COMP=HSAAM
```

This command is automatically issued when the last automation manager or the last automation agent in a single image terminates while the trace is active.

Before the trace records can be analyzed by IPCS, the trace data set or sets must be made available to IPCS. This is done with the operator command:

```
TRACE CT,WTRSTOP=HSACTWR
```

If the related message indicates that more than one data set contains trace data, you must use the IPCS command COPYTRC to combine the data sets into a new one.

To analyze the trace data, you must use the IPCS command CTRACE. A JCL is provided for the IPCS batch interface. The command supports several parameters and at the least, you must adapt the following:

- COMP(HSAAM)
- USEREXIT(HSAPSTUE)
- DSNAME(*trace_data_set_name*)
- OPTIONS((*filter_options*)), where *filter_options* is a combination of the options that are listed in Table 19 on page 220 and the following items, which are separated with a comma:

  CALLID=(*callid*[,*callid*])
  GROUP=*XCF_group_name*
  MAXTS=(yyyy-mm-dd[_hh:[:mm[:ss]]])
  MEMBER=(*XCF_group_member_name*[,*XCF_group_member_name*])
  METHOD=(*method*[,*method*])
  MINTS=(yyyy-mm-dd[_hh:[:mm[:ss]]])
  MODULE=(*module*[,*module*])
  TASKID=(*taskid*[,*taskid*])
  TOKEN=(*token*[,*token*])

  The keyword options are connected by a logical AND. The values of a keyword and the remaining options are connected by a logical OR. Apart from GROUP, MAXTS, and MINTS, each keyword option supports up to five different values that are separated by a comma or a space. Embedded spaces or commas in a value require delimiters, such as '...' or "...".

JCL Sample for Printing the Trace Records

The JCL sample member HSAJPTRC is stored in the SA z/OS SINGSAMP library. The data set names of the STEPLIB, IPCSDDIR, and TRACEDS DDnames must be changed to their actual values before the job is submitted. This is also true for all of the OPTIONS parameters that are listed because the values of the keywords are sample values only.

Using AOCTRACE

You can use the AOCTRACE command to enable or disable the automation debugging facility, either globally or for specific clists (REXX routines).

You can enable tracing for a particular message ID by issuing AOCTRACE at the command line with the MSG/*id* option, for example:

```
AOCTRACE MSG/AOF313I ON
```

You can find the syntax of the AOCTRACE command and additional information in *IBM Z System Automation Customizing and Programming*.

Information about using AOCTRACE to trace user-written automation procedures is provided in *IBM Z System Automation Customizing and Programming*.

For more information about trace settings, see TRACE in *TSO/E REXX/MVS Reference*.

How to Use a Log with INGHIST

With the INGHIST command you can display a log with automation manager messages by entering INGHIST REQ=LOG at the command line.

The INGHIST panel displays automation manager messages that are written to the system logger. Two or more lines are displayed for each message depending on the size of the message text. The first line contains the date and time when the message was issued. It also shows the XCF member name of

the automation manager and the XCF group name the automation manager belongs to. The second and following line, or lines, contain the message.

How to Use the Diagnostic Option of INGAMS

From the INGAMS command dialog, you can select option **D Diagnostic**.

This leads to the panel shown in Figure 55 on page 222 where you can initiate the following diagnostic functions:

- Writing a snapshot of the state image queue
- Starting or stopping recording
- Displaying work item statistics
- Starting or stopping work-item queue monitoring

```
INGKYAM4                  SA z/OS  - Command Dialogs
Domain ID   = IPUFJ     ---------- INGAMS   ----------    Date = 12/22/11
Operator ID = NETOPER1        Sysplex = AOC4PLEX          Time = 10:21:13

 Specify or revise the following data:

   Action  =>  __    1.  Write Snapshot
                     2.  Start recording
                     3.  Stop recording
                     4.  Work item Statistics
                     5.  Start queue monitoring - Interval => ___
                     6.  Stop queue monitoring

   Data set name  => ____________________________________________________________

   Target         => _____________ System name, domain id or sysplex name




Command ===>
   PF1=Help     PF2=End      PF3=Return                        PF6=Roll
                                                             PF12=Retrieve
```

Figure 55. INGAMS Diagnostic Functions Panel

In this panel's fields, enter the following information:

Action
Specify the action to be done. For options 1–3, you need to fill in the **Data set name** and **System** fields.

Enter 4 to display work item statistics. For details about enabling and using work item statistics, see “Work Item Statistics” on page 223.

Enter 5 and a value for the monitoring period (10–999 seconds) in the **Interval** field to activate work item queue monitoring. Enter 6 to stop queue monitoring. For more details, see “Workitem Queue Monitoring” on page 227.

Data set name
Specify the name of the data set that will hold the snapshot data.

The snapshot data set and the recording data set should be preallocated with RECFM FB and LRECL 80. The snapshot and recording data sets must be sequential files.

It is strongly recommended that you make your allocation of the snapshot data set according to the snapshot size information that you receive when you select option **B Show Details** for an automation manager from the INGAMS command dialog. This displays the information shown in Figure 56 on page 223 showing the snapshot size (highlighted in the example).

```
Diagnostic Info
    Snapshot size           : 3492224
    Number of resources     : 225
    Number of requests      : 66
    Number group requests   : 63
    History records         : 65422
    Max History records     : 262144
```

Figure 56. INGAMS Command Dialog, Show Details Panel

Note:

1. A data set name is required when writing a snapshot or when starting recording.
2. Make sure that the automation manager has the appropriate authority to write to the data set.

System
This is the name of the system where the automation manager resides. It is only required when the automation manager is not in the local sysplex. You can specify either the system name, the domain ID or the sysplex name.

Work Item Statistics

To display work item statistics, you enter 4 in the **Action** field of the INGAMS Diagnostic Functions Panel (Figure 55 on page 222) without specifying a data set name or system. This displays a panel similar to Figure 57 on page 223.

```
 INGKYAM5                  SA z/OS  - Command Dialogs      Line  1    of 11
Domain ID   = IPUFJ     ---------- INGAMS   ----------    Date = 12/22/11
Operator ID = NETOPER1        Sysplex = AOC4PLEX          Time = 16:46:03

Snapshot: 2011-12-22 16:46:01   Queue: 0           CPU time: 98.478

Cmd:  S Details
                       Since
Cmd Task       Status  (secs)  16:46  -1  -2  -3  -4  -5  -6  -7  -8  -9 -10
--- ---------- ------- ------- ----- --- --- --- --- --- --- --- --- --- ---
    ADMIN      IDLE
    QUERY1     IDLE                            1
    QUERY2     IDLE
    QUERY3     IDLE
    SCHED      IDLE                               1                   1
    UPDATER1   IDLE                               2                   2
    UPDATER2   IDLE
    UPDATER3   IDLE
    UPDATER4   IDLE
    ------------------------------------------------------------------------

Command ===>
   PF1=Help     PF2=End        PF3=Return                    PF6=Roll
                PF8=Forward    PF9=Refresh                 PF12=Retrieve
```

Figure 57. INGAMS Work Item Statistics Display

This panel shows history information about the work items processed by the automation manager. The automation manager keeps track of the last 300 work items processed by each of the tasks that build the automation manager kernel.

Note: The history information is not persistent or available when the automation manager runs on a system that is running an earlier version of SA z/OS.

The following data is shown:

Snapshot
The snapshot timestamp shows the time of the query.

Task
The name of the task.

Status
The status of the task. It is one of the following:

BUSY
The task is processing a work item.

IDLE
The task is waiting for work.

NOTIDLE
The task has completed work but is not waiting for work.

UNKNOWN
History records have not yet been defined for this task.

Since
This is the number of seconds that the task has been processing the work item (elapsed time). If this number is unexpectedly high, it is an indication that something is wrong.

The columns to the right of the **Since** column show the number of work items that have been processed in one minute, starting from the time the history query (INGAMS DIAG REQ=STATS) was made or refreshed.

You can use the S command code to display details of the work item history for the selected task, such as the starting time and total processing time of the work item, as shown in Figure 58 on page 224.

```
INGKYAM6                  SA z/OS  - Command Dialogs      Line  1    of 300
Domain ID   = IPSFO     ---------- INGAMS   ----------    Date = 11/23/09
Operator ID = NETOP1          Sysplex = KEY1PLEX          Time = 13:30:12

                                                Snapshot: 2009-11-23 13:29:55
               Starting Elapsed
 Task       Day   Time   in secs Busy Work item
 ---------- --- -------- ------- ---- ----------------------------------------
 QUERY1         11:53:36   0.043      GET_RESOURCE_DATA
 QUERY1         11:54:11   0.096      GET_RAW_RESOURCE_DATA
 QUERY1         11:54:11   0.067      GET_RAW_RESOURCE_DATA
 QUERY1         11:54:11   0.069      GET_RAW_RESOURCE_DATA
 QUERY1         11:54:12   0.067      GET_RAW_RESOURCE_DATA
 QUERY1         11:54:12   0.067      GET_RAW_RESOURCE_DATA
 QUERY1         11:54:13   0.067      GET_RESOURCE_DATA
 QUERY1         11:54:14   0.066      GET_RESOURCE_DATA
 QUERY1         11:54:15   0.067      GET_RESOURCE_DATA
 QUERY1         11:54:16   0.066      GET_RESOURCE_DATA
 QUERY1         11:54:17   0.067      GET_RESOURCE_DATA
 QUERY1         11:59:11   0.091      GET_RAW_RESOURCE_DATA

Command ===>
   PF1=Help     PF2=End        PF3=Return                    PF6=Roll
                PF8=Forward    PF9=Refresh                 PF12=Retrieve
```

Figure 58. INGAMS Work Item History Display

Work Item Lifecycle Recording

Work item lifecycle recording is an internal diagnostic tool that you should use only if required by SA z/OS service.

Lifecycle recording provides enhanced debugging to track down lost requests during automation agent-automation manager communication and other automation manager-related problems.

Lifecycle recording records checkpoints along the flow of a work item. The following flows are reported:

- Work items that flow from an automation agent to the primary automation manager (for example, status updates)
- Work items that flow from an automation agent to the primary automation manager plus the response (for example, INGLIST)
- Orders that flow from the PAM to an automation agent

Using Lifecycle Recording

By default, lifecycle recording is disabled. When enabled, the automation agent and the primary automation manager (PAM) write life cycle records to a data space. Each automation agent and the PAM have their own data space. Lifecycle recording must be enabled for the PAM and at least one automation agent. It can be enabled for the PAM from any automation agent, however it must be enabled for the automation agent from that automation agent.

Before lifecycle recording can be enabled, the size of the data space must be defined. This can be done automatically with the INGRLCR command that enables lifecycle recording.

When the problem has been captured, the data spaces must be off-loaded to external data sets of the same size with the INGRLCR command. A sequential data set is therefore required for the PAM and for each automation agent that lifecycle recording was enabled for.

The allocation of the data sets might be done automatically if the appropriate RACF definitions are given to the NetView that issues the lifecycle recording command.

Managing Lifecycle Recording with the INGRLCR Command

You can enable or disable lifecycle recording by issuing the following INGRLCR commands in NetView:

INGRLCR AA ON SIZE=*size* DSN=*dsname*
Switches on lifecycle recording for the local automation agent.

INGRLCR AA OFF
Switches off lifecycle recording for the local automation agent.

INGRLCR AM ON SIZE=*size* DSN=*dsname*
Switches on lifecycle recording for the PAM.

INGRLCR AM OFF
Switches off lifecycle recording for the PAM.

Note: If you issue the command with the size parameter, a new data space is created and the old data space is deleted. The dsname parameter specifies the data set that should be used later when the data space is saved. The total number of characters for the name of the data set is limited to 55 characters.

You can offload the data collected for lifecycle recording for the local automation agent or the primary automation manager, as follows:

- **Local automation agent:**
  - Use `INGRLCR AA SAVE DSN=`*dsname* if you used INGRLCR AA ON SIZE=*size* and did not specify a data set name or you want to save the data to a new data set.
  - Use `INGRLCR AA SAVE` if you used INGRLCR AA ON SIZE=*size* DSN=*dsname* to save the data to the data set that you specified.
- **Primary automation manager:**
  - Use `INGRLCR AM SAVE DSN=`*dsname* if you used INGRLCR AM ON SIZE=*size* and did not specify a data set name or you want to save the data to a new data set.
  - Use `INGRLCR AM SAVE` if you used INGRLCR AM ON SIZE=*size* DSN=*dsname* to save the data to the data set that you specified.

Notes:

1. When you offload the data space to a data set make sure that LCR is still ON before you issue SAVE.
2. Only the actual data from data space is copied. Small primary and multiple secondary allocations of the data set reduce the amount of DASD space.
3. If *dsname* does *not* exist it will be automatically created with the appropriate size. If the data set *already* exists it is used as it is. You should therefore make sure that it is large enough.

4. If *dsname* ends with *, for example, MY.HLQ.*, a new data set with a unique name is created using following pattern:

   ```
   MY.HLQ.<AA|AM>.<domain>.Dyymmdd.Thhmmss
   ```

5. NetView must have RACF ALTER access to the data set. The PAM must have RACF UPDATE access to the data set. Allocation of the new data set is done automatically via the NetView ALLOCATE command. The corresponding NetView return codes are passed back to caller in case of an error.
6. The size of the data space can range from 1–2098 MB. It specifies the maximum size of the data space. Initially a smaller data space is allocated that is extended until the maximum is reached. A value of 500 is recommended and should be sufficient for all situations.

Examples

```
INGRLCR AA ON SIZE=500 DSN=MY.AGENT.DATA.SET
INGRLCR AA ON SIZE=500 DSN=MY.AGENT.*
INGRLCR AA SAVE
INGRLCR AA SAVE DSN=MY.NEW.AGENT.DATA.SET
INGRLCR AA SAVE DSN=MY.HLQ.*
INGRLCR AM SAVE DSN=MY.NEW.PAM.DATA.SET WAIT=30
```

Changes to HSAPRMxx and INGXINIT

The LIFECYCLE parameter in the INGXINIT and HSAPRM*xx* members defines the size of the data space and the data set name for offloading the lifecycle recording data:

```
LIFECYCLE=nnnn;dsname
```

This parameter is not mandatory for using lifecycle recording. The same information can be provided later with the INGRLCR command when lifecycle recording is actually enabled.

Notes:

1. The LIFECYCLE parameter does not switch on lifecycle recording. This must be done with the INGRLCR AA|AM ON command.
2. The LIFECYCLE parameter does not allocate data sets automatically. This is done with the INGRLCR command only.
3. You must use a semicolon to separate *nnnn* and *dsname*. No blanks are allowed.

Performing Lifecycle Recording

Procedure

Carry out the following steps to perform lifecycle recording:

1. **Allocate data sets:**

   If necessary, allocate sequential data sets for the primary automation manager and each automation agent to be monitored with the following attributes:

   - Organization PS
   - Record format FB
   - Record length 128
   - Block size 27904

   The data sets must be large enough to hold the data space. The recommendation is 500 MB. The automation agents and PAM must have write access to their data sets.

2. **Create the data space:**

   After the automation agent has completed initialization, enable lifecycle recording on each automation agentthat is involved in debugging the problem with the following command:

   - INGRLCR AA ON SIZE=500 DSN=my.agent.dataset

Create the data space on the PAM just once with the following command:

- INGRLCR AM ON SIZE=500 DSN=my.pam.dataset

If the SIZE parameter is present, the old data space is deleted and a new empty data space created. The size and the data set name can also be provided with LIFECYCLE parameter in the HSAPRMxx and INGXINIT members.

3. **Check, enable or disable lifecycle**

   You can check the status of lifecycle recording with the following commands:

   - INGRLCR AA QRY
   - INGRLCR AM QRY

   If the data space has already been created (through either HSAPRMxx and INGXINIT, or INGRLCR), you can switch of lifecycle recording with the following commands, as needed:

   - INGRLCR AA ON
   - INGRLCR AM ON

   To switch off lifecycle recording, issue the following commands:

   - INGRLCR AA OFF
   - INGRLCR AM OFF

4. **Logging checkpoints:** When you have prepared and started lifecycle recording, recreate the problem situation. After the problem scenario has been duplicated, switch off lifecycle recording. Because the checkpoints are written in wraparound mode, do not unnecessarily delay switching off. This prevents important checkpoints from being lost or overwritten.
5. **Save data to data sets and send to SA z/OS service:**

   Use the following commands to copy the data space to the appropriate data sets that were previously allocated:

   - INGRLCR AA SAVE on each automation agentthat is involved in debugging the problem
   - INGRLCR AM SAVE

   Send the data sets to SA z/OS service for evaluation and analysis.

Displaying Lifecycle Recording Syntax

About this task

Issue the command INGRLCR HELP to display a short description of the syntax of the lifecycle recording command. It looks similar to the following output:

```
INGRLCR : IBM Diagnostic Tool for SA Life Cycle Recording
INGRLCR : target: HELP function:
INGRLCR : extracted: size= dsn= wait=
INGRLCR : SYNTAX
INGRLCR : PARAMETERS: target function KEY=VALUE [ALLOC-parms]
INGRLCR : target    : AA|AM
INGRLCR : function  : SAVE|ON|OFF|QRY|FLUSH
INGRLCR : KEY=VALUE : SIZE=nn  DSN=a.b.c  WAIT=sec
INGRLCR : ALLOC-parms might be optional NetView ALLOC parameters
INGRLCR : Build Date: 19 Nov 2009
INGRLCR : DONE with return code 0
```

Workitem Queue Monitoring

Timeout messages (ING008I) can be the result of the fact that more work items are received than the automation manager is able to process. To better understand the nature of received timeout messages, you can now monitor the workitem input queue. This can be achieved with the INGAMS STARTMON/STOPMON parameter.

When the automation manager workitem queue monitoring routine detects that more work items are arriving than can be processed within the monitoring interval, message INGX1011I is issued.

In addition, INGRYAMD traps message INGX1011I with the NetView automation table and saves the relevant data. On request, INGRYAMD creates a file that contains the collected data. The data is separated by semicolons so that the file can be easily imported into a spreadsheet.

The syntax of the INGRYAMD command is as follows:

REQ
Specifies the function to be performed. Valid values are:

REPORT
Generates a report file that contains the captured data.

RESET
Resets the captured data.

OUTDSN
Specifies the name of the data set that contains the report file. The data set must exist. The minimum record length should be 80.

The report file contains one record for each message that is captured. The format of the records is:

Column #	Description
1	Timestamp when INGX1011I message was captured. The format is yyyy-mm-dd hh:mm:ss
2	(DIF) The number of work items that were not processed within the last monitoring cycle. It is the number of work items in the queue at the current monitoring time minus the number of work items in the queue at the last monitoring period.
3	(ABS) The total number of work items in the input and output queue. This also includes the work items that have been processed but not yet run through the cleanup phase.
4	(QUE) The number of work items in the input queue.
5	(EXP) The number of expired work items in the input queue.
6	(CPU) The CPU time of the automation manager.
7	(OUT) The number of entries in the output queue.
8	(RSP) The number of response entries in the input queue (for example, Query).
9	(FAF) The number of fire and forget entries in the input queue.

Tracing and Debugging for ProcOps and the BCP Internal Interface

The following trace facilities are provided for ProcOps SNMP connections and the BCP Internal Interface that you can use for problem determination purposes.

- "Using AOCTRACE" on page 228
- "Tracing Specific Connections" on page 229
- "First Failure Data Capture Support" on page 230

Using AOCTRACE

To enable the global tracing of all communications related parts of ProcOps-SNMP or BCP Internal Interface, issue the following command:

```
AOCTRACE compname ON
```

Where *compname* is one of the following:

INGRX810
INGHWCMD command list

INGHWCOM
Communications module

To disable the global trace mode issue command:

```
AOCTRACE compname OFF
```

In global trace mode all currently active connections to SEs or HMCs are traced. This may produce a large amount of data written to the netlog, depending on the number of connections or the amount event data returned from the hardware. In such cases, use the connection-specific trace, as described in "Tracing Specific Connections" on page 229.

Note: For AOCTRACE INGHWCOM, it is important to consider the target hardware initialization from a time point of view as well. Keep in mind that a target hardware connection deals with two different sessions:

1. Event session - established with command INITCOM for connection type INTERNAL (BCPii) or command ISQXIII for connection type SNMP (Processor Operations); the event session remains active till the connection closed with command TERMCOM (BCPii) or ISQXCLS (Processor Operations). The event session handles hardware events and system messages.
2. User session - established for each distinct command issued by the user and terminated after command execution.

In order to get trace messages for BOTH sessions, command AOCTRACE INGHWCOM ON needs to be issued BEFORE command INITCOM or ISQXIII is executed. To stop AOCTRACE, two steps are required:

1. Issue AOCTRACE INGHWCOM OFF.
2. Stop any active target hardware connection (using command TERMCOM or ISQXCLS). If this step is not done, AOCTRACE messages are stopped only for User Command sessions, but will still be issued for Event sessions.

After target hardware connection stop completion, the stopped connections can be restarted immediately. If AOCTRACE INGHWCOM is issued AFTER target hardware connection initialization, trace messages will be written for User Command sessions. To stop tracing, issue command AOCTRACE INGHWCOM OFF.

Tracing Specific Connections

If you want to trace the communication to a specific processor or CPC, issue one of the following service commands:

```
INGHWCMD proname TRACE ON
INGHWCMD proname UTRACE ON
```

Where *proname* is the name of the target hardware or processor that is defined in the SA z/OS customization dialog, and specifying UTRACE traces only the user session of the operator that issued the command.

Use the following service commands to disable the trace:

```
INGHWCMD proname TRACE OFF
INGHWCMD proname UTRACE OFF
```

Although only a single processor connection is traced, the amount of data that is written to the netlog may be high, especially if the processor connection is through an HMC. In this case, data from other processors that are serviced by this HMC is additionally shown in the trace. It is therefore recommended to limit the tracing time.

First Failure Data Capture Support

The INGHWCOM communications module supports the First Failure Data Capture debug aid, provided by NetView for high level language command processors, written in PL/I or C. In case of an INGHWCOM abnormal end, message CNM983E, CNM998E, or CNM999E is issued.

Message CNM983E contains the HLL API user trace information that is maintained by INGHWCOM. This information allows you to determine the internal program flow at error time. If available, save this debug aid information before contacting IBM for service. For more information about the diagnostic messages, see *IBM Z NetView Messages and Codes*.

Collecting the available BCPII session status messages

In case of BCPii session problems, make sure you have collected all SA z/OS WTO messages (msgids ING81*) at the time of the error, that are issued in the SA z/OS NetView domain, where the error occurred. Report these messages (Syslog,Netlog) in case you need to contact IBM Support.

The BCPii session status messages of interest are: ING811I, ING812I, ING813E, ING814E, ING815I, ING816I, ING817I, ING818I and ING819I.

Appendix D. Hardware Interfaces: Additional Recovery Information

BCP Internal Interface Recovery Information

BCP Internal Interface Recovery Actions

The following section explains manual recovery procedures that might be necessary in case the BCP internal interface communication module INGHWCOM is not responding or is unavailable.

Interface report messages that indicate that module INGHWCOM is unavailable are:

1. AOFA0000 Check Task: <*tn*>_task_module_INGHWCOM_not_running
2. AOFA0000 Check Task: <*tn*>_reached QueueLimit_<*ql*>
3. AOFA0000 Check Task: <*tn*>_task_is_not_available

The recovery procedure for each of these cases is outlined below.

Recovery Case 1: <*tn*>_task_module_INGHWCOM_not_running

About this task

The recovery procedure for this case is as follows:

Procedure

1. Make a note of the autotask name <*tn*> and browse the netlog for messages to or from that task.
2. Search for DSI172I messages that have <*tn*> as the abending subtask and look for INGHWCOM as the failing module. If this is the case, note down the last report message (AOFA*) prior to the abend from that task.
3. Issue the NetView command EXCMD <*tn*> LOGOFF (where *tn* is the autotask name that you made a note of in step 1).

   This will terminate the autotask and free its task storage that was allocated by INGHWCOM. If the autotask has been defined as an HWOPERnn task in the SA z/OS AUTOOPS policy entry, it is automatically restarted and activates the communication module INGHWCOM.

Recovery Case 2: <*tn*>_reached QueueLimit_<*ql*>

Each Support Element connection uses a separate autotask. As soon as the "reached QueueLimit" message is returned, this indicates that the previously issued HW request has not yet sent a completion report. The normal recovery process would be to retry the HW command request at a later time.

There are, however, cases where module INGHWCOM has to wait for an indefinite period of time until the HW has completed the request. Usually, this occurs when a Support Element session has been started in SYNC processing mode while a long-running HW command such as ACTIVATE or CBU is still active.

If necessary, you can interrupt and terminate the session, which will not affect any HW command currently in progress on the Support Element. If you terminate the session, you will not receive any completion information for this command.

Issue the following NetView commands:

1. EXCMD <*tn*> RESET
2. TASKUTIL <*tn*>

3. EXCMD <*tn*> LOGOFF

Command "1" on page 231 terminates the communication module INGHWCOM in task <*tn*>. The session between NetView and the SE/HMC will be closed normally and the allocated task storage will be freed.

With command "2" on page 231 you can control whether module INGHWCOM is no longer the active command of <*tn*> and the amount of storage still allocated by <*tn*>.

With command "3" on page 232 you terminate autotask <*tn*>, which will initiate an autotask restart and the activation of INGHWCOM.

If command "2" on page 231 indicates that command "1" on page 231 did not end INGHWCOM, command "1" on page 231 can be repeated with RESET IMMED. This will abnormally terminate INGHWCOM, but SE/HMC session termination will not be performed. Note that a 'STOP TASK' command will end the autotask, but will not invoke normal SE/HMC session termination.

Recovery Case 3: <*tn*>_task_is_not_available

About this task

There are cases where a BCP internal interface autotask, previously defined as an HWOPER*nn* task with an SA z/OS AUTOOPS policy entry, cannot be used. In these cases, a task definition problem or a mismatch of your current ACF configuration might be the reason for the problem.

The recovery procedure for this case is as follows:

Procedure

1. Verify the AUTOOPS definition statements in your PDB.
2. Verify that the corresponding DSIOPF statements have AOFPRFHW defined as the operator profile name.
3. Perform a Build operation after the policy changes.
4. Activate your policy changes, for example by performing an ACF REFRESH.

Checking BCP Internal Interface Interlocks

If applications that exploit the BCP Internal Interface, such as GDPS®, seem to hang or loop while a HW command or HW query is active, this may indicate that internal locks were not freed. Unconditional task interrupts during INGHWCMD processing or an unconditional task termination can cause such a situation.

You can use the LCHECK function of the IBM Service command INGHWSRV to clear blocking locks. For more information about the service command, use the SA z/OS command HELP INGHWSRV.

Terminating NetView and the BCP Internal Interface

When terminating NetView, a 'CLOSE NORMAL' command should first be issued to give INGHWCOM enough time to end all its SE/HMC sessions. Then, after a predefined period of time, a 'CLOSE IMMED' command could be used to limit the overall termination time for NetView.

How to Switch the IP Address for Target Hardware Used by Processor Operations

This description is only valid for target hardware (defined as 'Processors' in the SA z/OS customization dialog) with a connection type of SNMP.

When defining a processor as the target hardware for Processor Operations, two IP addresses can be provided for communication with either the Support Element (SE) or the Hardware Management Console (HMC). At least one IP address for the SE or HMC must be provided; optionally, a secondary IP address can be provided for this processor. If two IP addresses are provided and both addresses are SE ones, ensure that the SE has an alternate network adapter card installed and that it is operational.

If you need to switch between these two addresses (for example, in case of a SE network adapter card failure), switching to the secondary IP address can be easily performed using Processor Operations command:

```
ISQIPSWT <target-hardware-name>
```

For a detailed command description, refer to *IBM Z System Automation Customizing and Programming*.

How to Terminate BCP Internal Interface Connections or Clear Hardware Control Blocks after a Connection Failure

There are occasions when it may be necessary to terminate BCP internal interface connections, for example, prior to issuing a CLOSE IMMED to terminate Z NetView and Z System Automation.

This can be done using the following Systems Operations command:

```
INGHWSRV TERM
```

This command performs a TERMCOM request for each CPC that is found and then attempts to disable the HSAET32 resource.

Occasionally it may be necessary to clear hardware control blocks after a connection failure with the target hardware.

This can be done using the following Systems Operations command:

```
INGHWSRV target_hardware_name FORCE
```

For a detailed command description, see *IBM Z System Automation Operator's Commands*.

See also the appendix "Controlling Access to the Processor Hardware Functions" in *IBM Z System Automation Planning and Installation*.

Recovery of Processor Operations Target System Connections in the event of Hardware Management Console Outage

Processor Operations target system connections rely on a fully operational Support Element (SE) or a Hardware Management Console (HMC), in order to monitor active target system connections or to execute commands against defined target systems.

If the connection to the HMC is lost, there are 2 major reasons for the failure:

1. Shutdown of the HMC for maintenance reasons (planned outage).
2. Technical failure or accidental shutdown of the HMC (unplanned outage).

In case of a planned outage, it is strongly recommended to close any active target system connections for that HMC as follows:

1. Issue Processor Operations host-based command ISQXCLS, or
2. Issue Line command 'C' in the Processor Operations target system summary panel (see host-based command ISQXDST), or
3. Issue Processor Operations common command CTRLCONS (see the note at the end of this section regarding command CTRLCONS).

Once the HMC is active again, the previously closed target system connections can be re-initiated using host-based command ISQXIII , or line command 'I' in the Processor Operations target system summary panel (see host-based command ISQXDST).

For an unplanned outage, check on the duration of the outage and if Processor Operations has received an error notification (event message from the HMC).

If the interrupt is for a short period, and if Processor Operations receives an event failure from the HMC (messages AOFA0998 and AOFA0999 as described in *IBM Z System Automation Messages and Codes*), Processor Operations might succeed in re-establishing the broken connection on its own immediately.

If Processor Operations does not receive any event failure message, the connection will remain in an INITIALIZED status, though it may change to, for example, TARGET HARDWARE PROBLEM if the status polling process detects there is a problem. If the target system connection has this problem status, AND if the reason obviously is due to an outage of the HMC, AND if Processor Operations did not succeed in automatically re-establishing the target system connection, the recommended recovery steps are as follows:

1. Wait till the affected HMC is available again,
2. Issue command ISQXCLS for all affected target systems to properly close and clean up the Processor Operations connection,
3. Issue command ISQXIII for all affected target systems to re-establish Processor Operations connection.

Note: In order to shutdown the HMC in a controlled manner, the Processor Operations common command CTRLCONS can be used (see *IBM Z System Automation Customizing and Programming* for a detailed description). CTRLCONS will automatically close any still active target system connections to that HMC.

Using the ProcOps HOLD Session Mode

The ProcOps HOLD session mode is available with active ISQET32 sessions only. You can put an ISQET32 session in the HOLD mode for the following purposes:

- Disable the usage and automation of an active ISQET32 session.
- Make a session unavailable for a planned Support Element outage, as an alternative to SUSPEND the session.

When an ISQET32 session is in HOLD mode, any messages that are received from hardware event or operating system are prefixed with an AOFA9999 message token and written to the netlog. No automation is triggered and the CPC or image status information is not updated. Except **ISQXCON RESUME**, other session management commands are not accepted. You cannot send ISQCCMD common commands to a target hardware or a target system when the related session is in HOLD mode. If you want to stop ProcOps by using the ISQSTOP command, sessions in HOLD mode are terminated. In contrast to sessions in SUSPENDED mode, sessions in HOLD mode do not keep this status across a NetView or SA recycle as HOLD is a transient logical session mode only.

You can put a session in HOLD mode manually by either issuing the **ISQXCON *thwname* SUSPEND HOLD** command, or by using the **ISQXDST** status dialog. On the status dialog, select a target system, type H in the command field and press the Enter key. As a response, the target status in the **ISQESUM** panel changes to HOLD for the selected system and all other target systems that use the same ISQET32 session to the Support Element.

You can resume a session from HOLD mode manually by either issuing the **ISQXCON *thwname* RESUME** command, or by using the **ISQXDST** status dialog. On the status dialog, select a target system, type R in the command field and press the Enter key. As a response, the target status in the **ISQESUM** panel changes to INITIALIZED for the selected system and all other target systems that use the same ISQET32 session to the Support Element. If no connection to the Support Element is possible at the time of the resume, the status changes either to CLOSED or SNMP SESSION PROBLEM, depending on the error situation.

You can use the **ISQXCON * STATUS** command to display the current session status for all configured ProcOps sessions.

Appendix E. Processor Operations – setup for dynamic target system names

This topic describes how to use the AOF_AAO_ISQ_DYNTGT variable to define a dynamic target systems name, without the need to define a corresponding system entry and a "dummy" target system name.

The Processor Operations data model requires a distinct target system name for each LPAR/image. The LPARs and target system names are defined in the Customization Dialog > Processors entry > LPARS AND SYSTEMS policy.

Here is a sample.

```
                      LPAR Definitions

Entry Type : Processor             PolicyDB Name   : TESTPDB
Entry Name : HEL2                  Enterprise Name : MYENTERPRISE

  Action     LPAR Name   Target Mode   Target System Name
  _________  KEY1_____   ESA_____      _________
  _________  KEY2_____   ESA_____      _________
  _________  KEY3_____   ESA_____      KEY3_____
  _________  KEY4_____   ESA_____      KEY4_____
```

In this sample, LPARs KEY3 and KEY4 have Target System Name defined, while KEY1 and KEY2 not. If a Target System Name is defined here, a corresponding system entry with the same name must exist.

When Processor Operations is started, the standard procedure is to process only the defined LPARs that have Target System Names assigned. LPARs without Target System Names are skipped from being loaded into the Processor Operations data model.

For a specific LPAR, it is a common usage to define a backup LPAR with the same name on a different CPC. As Target System Names must be unique, you cannot assign that name twice, but need to define a "dummy" system name (for example, KEY3BKUP for the sample above).

Now you can avoid the additional effort to define "dummy" system names. You can use the advanced automation global variable AOF_AAO_ISQ_DYNTGT in the NetView stylesheet to dynamically create the Target System Names at Processor Operations start time.

The pattern value of AOF_AAO_ISQ_DYNTGT can be composed from the following 3 building blocks:

- L*x*: use LPAR name from 1 to *x*.
- C*y*: use PROCESSOR name from 1 to *y*.
- Special characters '#$@' (only one is allowed.)

Some samples for Processor **HEL2** and LPAR **KEY2**:

Sample	Meaning	Composed target system name
L8	Take all 8 characters of the LPAR name.	KEY2
L7#	Take the first 7 characters of the LPAR name and append special character #.	KEY2#
L4C3@	Take the first 4 characters of the LPAR name and the first 3 characters of the Processor name, and append special character @.	KEY2HEL@

Note that the maximum length of a target system name is 8 characters.

Example

Here is an example, showing how to create the dynamic target system names.

1. Define LPARs. For example, LPARs KEY3 and KEY4 in Processor HEL2, with Target System Names defined.

```
                     LPAR Definitions

Entry Type : Processor             PolicyDB Name   : TESTPDB
Entry Name : HEL2                  Enterprise Name : MYENTERPRISE

  Action     LPAR Name   Target Mode   Target System Name
  _________  KEY3_____   ESA_____      KEY3_____
  _________  KEY4_____   ESA_____      KEY4_____
```

2. Define backup LPARs. For example, KEY3 and KEY4 in Processor BKP2. Leave the Target System Names blank.

```
                     LPAR Definitions

Entry Type : Processor             PolicyDB Name   : TESTPDB
Entry Name : BKP2                  Enterprise Name : MYENTERPRISE

  Action     LPAR Name   Target Mode   Target System Name
  _________  KEY3_____   ESA_____      __________
  _________  KEY4_____   ESA_____      __________
```

3. Set the global variable AOF_AAO_ISQ_DYNTGT in the NetView stylesheet. For example, "L4#".

```
*****************************************************
* System Automation AAO CGlobals
*****************************************************
. . .
COMMON.AOF_AAO_ISQ_DYNTGT = L4#
```

4. Start the Processor Operations using the ISQSTART command. LPARs with Target System Names are loaded into the Processor Operations data model.

   Note: For dynamic target systems, there is no system entry data available. 4 data values are defaulted.

   - Operating System Type: set to value “MVS"
   - Initialize target system: set to value “BYPASS”
   - Default Response to IEA213A/IEA214A: set to value “YES”
   - Target time zone: set to value “0” (i.e. as same as host)

5. Use ISQXDST command to display the target systems. KEY3# and KEY4#, as shown below, are the created dynamic target systems.

```
ISQESUM    SA z/OS Proc-Ops Target Status Summary    Updates: Dynamic
   Configuration in use: ACF

I isqxiii C isqxcls O isqxopt  A,B view netlogs E events   Debug Mode: OFF
S suspend /  H hold THW path       R Resume a suspended / hold THW path
Cmd    Target System   Status      Focal Points - Primary: PRIM Backup: BKUP
       KEY3            CLOSED
       KEY4            CLOSED
       KEY3#           CLOSED
       KEY4#           CLOSED
```

Appendix F. Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Glossary

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing* New York: McGraw-Hill, 1994.
- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Deprecated term for. This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

See. This refers the reader to multiple-word terms in which this term appears.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

A

ACF
See automation configuration file.

ACF/NCP
Advanced Communications Function for the Network Control Program. See Advanced Communications Function and Network Control Program.

ACF/VTAM
Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for VTAM. See Advanced Communications Function and Virtual Telecommunications Access Method.

active monitoring
In SA z/OSautomation control file, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also passive monitoring.

adapter
Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

adjacent hosts
Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

adjacent NetView
In SA z/OS, the system defined as the communication path between two SA z/OS systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA z/OS systems. For example, the adjacent NetView is used when sending responses from a focal point to a remote system.

Advanced Communications Function (ACF)
A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

advanced program-to-program communication (APPC)
A set of inter-program communication services that support cooperative transaction processing in a Systems Network Architecture (SNA) network. APPC is the implementation, on a given system, of SNA's logical unit type 6.2.

Advanced Workload Analysis Reporter (zAware)
IBM analytics appliance running in a z Systems® partition, activated in zACI mode. Customers can use the appliance to monitor the console message streams of other LPARs running in the same IBM Z cluster and create trend reports. Exploiting zAware and these trend reports can help to better predict OS outages or performance degradations and initiate proactive clusters.

alert
In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition.

In NetView, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

alert condition
A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.

alert threshold
An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the alert color. SA z/OS may also issue an alert. See warning threshold.

AMC
See Automation Manager Configuration.

American Standard Code for Information Interchange (ASCII)
A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters (8-bit including parity check). The ASCII set consists of control characters and graphic characters. See also Extended Binary Coded Decimal Interchange Code.

APF
See authorized program facility.

API
See application programming interface.

APPC
See advanced program-to-program communication.

application
In SA z/OS, applications refer to z/OS subsystems, started tasks, or jobs that are automated and monitored by SA z/OS. On SNMP-capable processors, application can be used to refer to a subsystem or process.

Application entry
A construct, created with the customization dialogs, used to represent and contain policy for an application.

application group
A named set of applications. An application group is part of an SA z/OS enterprise definition and is used for monitoring purposes.

application program
A program written for or by a user that applies to the user's work, such as a program that does inventory or payroll.

A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application programming interface (API)
An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

ApplicationGroup entry
A construct, created with the customization dialogs, used to represent and contain policy for an application group.

ARM
See automatic restart management.

ASCB
Address space control block.

ASCB status
An application status derived by SA z/OS running a routine (the ASCB checker) that searches the z/OS address space control blocks (ASCBs) for address spaces with a particular job name. The job name used by the ASCB checker is the job name defined in the customization dialog for the application.

ASCII
See American Standard Code for Information Interchange.

ASF
See automation status file.

authorized program facility (APF)
A facility that permits identification of programs that are authorized to use restricted functions.

automated console operations (ACO)
The use of an automated procedure to replace or simplify the action that an operator takes from a console in response to system or network events.

automated function
SA z/OS automated functions are automation operators, NetView autotasks that are assigned to perform specific automation functions. However, SA z/OS defines its own synonyms, or *automated function names*, for the NetView autotasks, and these function names are referred to in the sample policy databases provided by SA z/OS. For example, the automation operator AUTBASE corresponds to the SA z/OS automated function BASEOPER.

automatic restart management (ARM)
A z/OS recovery function that improves the availability of specified subsystems and applications by automatically restarting them under certain circumstances. Automatic restart management is a function of the Cross-System Coupling Facility (XCF) component of z/OS.

automatic restart management element name
In MVS 5.2 or later, z/OS automatic restart management requires the specification of a unique sixteen character name for each address space that registers with it. All automatic restart management policy is defined in terms of the element name, including the SA z/OS interface with it.

automation
The automatic initiation of actions in response to detected conditions or events. SA z/OS provides automation for z/OS applications, z/OS components, and remote systems that run z/OS. SA z/OS also provides tools that can be used to develop additional automation.

automation agent
In SA z/OS, the automation function is split up between the automation manager and the automation agents. The observing, reacting and doing parts are located within the NetView address space, and are known as the *automation agents*. The automation agents are responsible for:

- Recovery processing
- Message processing
- Active monitoring: they propagate status changes to the automation manager

automation configuration file
The SA z/OS customization dialogs must be used to build the automation configuration file. It consists of:

- The automation manager configuration file (AMC)
- The NetView automation table (AT)
- The NetView message revision table (MRT)
- The MPFLSTxx member

automation control file (ACF)
In SA z/OS, a file that contains system-level automation policy information. There is one master automation control file for each NetView system that SA z/OS is installed on. Additional policy information and all resource status information is contained in the policy database (PDB). The SA z/OS customization dialogs must be used to build the automation control files. They must not be edited manually.

automation flags
In SA z/OS, the automation policy settings that determine the operator functions that are automated for a resource and the times during which automation is active. When SA z/OS is running, automation is controlled by automation flag policy settings and override settings (if any) entered by the operator. Automation flags are set using the customization dialogs.

automation manager
In SA z/OS, the automation function is split up between the automation manager and the automation agents. The coordination, decision making and controlling functions are processed by each sysplex's ***automation manager***.

The automation manager contains a model of all of the automated resources within the sysplex. The automation agents feed the automation manager with status information and perform the actions that the automation manager tells them to.

The automation manager provides ***sysplex-wide*** automation.

Automation Manager Configuration
The Automation Manager Configuration file (AMC) contains an image of the automated systems in a sysplex or of a standalone system. See also automation configuration file.

Automation NetView
In SA z/OS the NetView that performs routine operator tasks with command procedures or uses other ways of automating system and network management, issuing automatic responses to messages and management services units.

automation operator
NetView automation operators are NetView autotasks that are assigned to perform specific automation functions. See also automated function. NetView automation operators may receive messages and process automation procedures. There are no logged-on users associated with automation operators. Each automation operator is an operating system task and runs concurrently with other NetView tasks. An automation operator could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the automation operator. Similar to *operator station task*. SA z/OS message monitor tasks and target control tasks are automation operators.

automation policy
The policy information governing automation for individual systems. This includes automation for applications, z/OS subsystems, z/OS data sets, and z/OS components.

automation policy settings
The automation policy information contained in the automation control file. This information is entered using the customization dialogs. You can display or modify these settings using the customization dialogs.

automation procedure
A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under NetView.

automation routines
In SA z/OS, a set of self-contained automation routines that can be called from the NetView automation table, or from user-written automation procedures.

automation status file (ASF)
In SA z/OS, a file containing status information for each automated subsystem, component or data set. This information is used by SA z/OS automation when taking action or when determining what action to take. In Release 2 and above of AOC/MVS, status information is also maintained in the operational information base.

automation table (AT)
See NetView automation table.

autotask
A NetView automation task that receives messages and processes automation procedures. There are no logged-on users associated with autotasks. Each autotask is an operating system task and runs concurrently with other NetView tasks. An autotask could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the autotasks. Similar to *operator station task*. SA z/OS message monitor tasks and target control tasks are autotasks. Also called *automation operator*.

available
In VTAM programs, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

B

Base Control Program (BCP)
A program that provides essential services for the MVS and z/OS operating systems. The program includes functions that manage system resources. These functions include input/output, dispatch units of work, and the z/OS UNIX System Services kernel. See also Multiple Virtual Storage and z/OS.

basic mode
A central processor mode that does not use logical partitioning. Contrast with logically partitioned mode.

BCP
See Base Control Program.

BCP Internal Interface
Processor function of IBM Z processor families. It allows for communication between basic control programs such as z/OS and the processor support element in order to exchange information or to perform processor control functions. Programs using this function can perform hardware operations such as ACTIVATE or SYSTEM RESET.

beaconing
The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

BookManager®
An IBM product that lets users view softcopy documents on their workstations.

C

central processor (CP)
The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

central processor complex (CPC)
A physical collection of hardware that consists of central storage, (one or more) central processors, (one or more) timers, and (one or more) channels.

central site
In a distributed data processing network, the central site is usually defined as the focal point for alerts, application design, and remote system management tasks such as problem management.

channel
A path along which signals can be sent; for example, data channel, output channel. See also link.

channel path identifier
A system-unique value assigned to each channel path.

channel-attached
Attached directly by I/O channels to a host processor (for example, a channel-attached device).

Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with link-attached. Synonymous with local.

CHPID
In SA z/OS, channel path ID; the address of a channel.

CHPID port
A label that describes the system name, logical partitions, and channel paths.

CI
See console integration.

CICS/VS
Customer Information Control System for Virtual Storage. See Customer Information Control System.

CLIST
See command list.

clone
A set of definitions for application instances that are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

clone ID
A generic means of handling system-specific values such as the MVS SYSCLONE or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

command
A request for the performance of an operation or the execution of a particular program.

command facility
The component of NetView that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

command list (CLIST)
A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled.

In SA z/OS, REXX command lists that can be used for automation procedures.

command procedure
In NetView, either a command list or a command processor.

command processor
A module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are issued as commands.

Command Tree/2
An OS/2-based program that helps you build commands on an OS/2 window, then routes the commands to the destination you specify (such as a 3270 session, a file, a command line, or an application program). It provides the capability for operators to build commands and route them to a specified destination.

common commands
The SA z/OS subset of the CPC operations management commands.

Common User Access (CUA) architecture
Guidelines for the dialog between a human and a workstation or terminal.

communication controller
A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit or by a program executed in a processor to which the controller is connected. It manages the details of line control and the routing of data through a network.

communication line
Deprecated term for telecommunication line.

connectivity view
In SA z/OS, a display that uses graphic images for I/O devices and lines to show how they are connected.

console automation
The process of having NetView facilities provide the console input usually handled by the operator.

console connection
In SA z/OS, the 3270 or ASCII (serial) connection between a PS/2 computer and a target system. Through this connection, the workstation appears (to the target system) to be a console.

console integration (CI)
A hardware facility that if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.

consoles
Workstations and 3270-type devices that manage your enterprise.

couple data set
A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also sysplex couple data setand XCF couple data set.

coupling facility
The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

CP
See central processor.

CPC
See central processor complex.

CPC operations management commands
A set of commands and responses for controlling the operation of System/390® CPCs.

CPC subset
All or part of a CPC. It contains the minimum *resource* to support a single control program.

CPU
Central processing unit. Deprecated term for processor.

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

Customer Information Control System (CICS)
A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

customization dialogs
The customization dialogs are an ISPF application. They are used to customize the enterprise policy, like, for example, the enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise. How to use these dialogs is described in *IBM Z System Automation Customizing and Programming*.

D

DataPower® X150z
See IBM Websphere DataPower Integration Appliance X150 for zEnterprise® (DataPower X150z).

DASD
See direct access storage device.

data services task (DST)
The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

data set
The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set members
Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

DBCS
See double-byte character set.

DCCF
See disabled console communication facility.

DCF
See Document Composition Facility.

DELAY Report
An RMF report that shows the activity of each job in the system and the hardware and software resources that are delaying each job.

device
A piece of equipment. Devices can be workstations, printers, disk drives, tape units, remote systems or communications controllers. You can see information about all devices attached to a particular switch, and control paths and jobs to devices.

DEVR Report
An RMF report that presents information about the activity of I/O devices that are delaying jobs.

dialog
Interactive 3270 panels.

direct access storage device (DASD)
A device that allows storage to be directly accessed, such as a disk drive.

disabled console communication facility (DCCF)
A z/OS component that provides limited-function console communication during system recovery situations.

disk operating system (DOS)
An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

Software for a personal computer that controls the processing of programs. For the IBM Personal Computer, the full name is Personal Computer Disk Operating System (PCDOS).

display
To present information for viewing, usually on the screen of a workstation or on a hardcopy device.

Deprecated term for panel.

distribution manager
The component of the NetView program that enables the host system to use, send, and delete files and programs in a network of computers.

Document Composition Facility (DCF)
An IBM licensed program used to format input to a printer.

domain
An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations.

In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and associated resources that the SSCP can control with activation requests and deactivation requests.

double-byte character set (DBCS)
A character set, such as Kanji, in which each character is represented by a 2-byte code.

DP enterprise
Data processing enterprise.

DSIPARM
This file is a collection of members for NetView customization.

DST
Data Services Task.

E

EBCDIC
See Extended Binary Coded Decimal Interchange Code.

ECB
See event control block.

EMCS
Extended multiple console support. See also multiple console support.

enterprise
The composite of all operational entities, functions, and resources that form the total business concern and that require an information system.

Enterprise Systems Architecture (ESA)
A hardware architecture that reduces the effort required for managing data sets and extends addressability for system, subsystem, and application functions.

entries
Resources, such as processors, entered on panels.

entry type
Resources, such as processors or applications, used for automation and monitoring.

environment
Data processing enterprise.

error threshold
An automation policy setting that specifies when SA z/OS should stop trying to restart or recover an application, subsystem or component, or offload a data set.

ESA
See Enterprise Systems Architecture.

event
In NetView, a record indicating irregularities of operation in physical elements of a network.

An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation.

Events are part of a trigger condition, such that if all events of a trigger condition have occurred, a startup or shutdown of an application is performed.

event control block (ECB)
A control block used to represent the status of an event.

exception condition
An occurrence on a system that is a deviation from normal operation. SA z/OS monitoring highlights exception conditions and allows an SA z/OS enterprise to be managed by exception.

Extended Binary Coded Decimal Interchange Code (EBCDIC)
A coded character set of 256 8-bit characters developed for the representation of textual data. See also American Standard Code for Information Interchange.

extended recovery facility (XRF)
A facility that minimizes the effect of failures in z/OS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.

F

fallback system
See secondary system.

field
A collection of bytes within a record that are logically related and are processed as a unit.

file manager commands
A set of SA z/OS commands that read data from or write data to the automation control file or the operational information base. These commands are useful in the development of automation that uses SA z/OS facilities.

focal point
In NetView, the focal-point domain is the central host domain. It is the central control point for any management services element containing control of the network management data.

focal point system
A system that can administer, manage, or control one or more target systems. There are a number of different focal point system associated with IBM automation products.

SA z/OS Processor Operations focal point system. This is a NetView system that has SA z/OS host code installed. The SA z/OS Processor Operations focal point system receives messages from the systems and operator consoles of the machines that it controls. It provides full systems and operations console function for its target systems. It can be used to IPL these systems. Note that some restrictions apply to the Hardware Management Console for an S/390® microprocessor cluster.

SA z/OS SDF focal point system. The SA z/OS SDF focal point system is an SA z/OS NetView system that collects status information from other SA z/OS NetViews within your enterprise.

Status focal point system. In NetView, the system to which STATMON, VTAM and NLDM send status information on network resources.

Hardware Management Console. Although not listed as a focal point, the Hardware Management Console acts as a focal point for the console functions of an S/390 microprocessor cluster. Unlike all the other focal points in this definition, the Hardware Management Console runs on a LAN-connected workstation,

frame
For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

full-screen mode
In NetView, a form of panel presentation that makes it possible to display the contents of an entire workstation screen at once. Full-screen mode can be used for fill-in-the-blanks prompting. Contrast with line mode.

G

gateway session
An NetView-NetView Task session with another system in which the SA z/OS outbound gateway operator logs onto the other NetView session without human operator intervention. Each end of a gateway session has both an inbound and outbound gateway operator.

generic alert
Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as NetView.

group
A collection of target systems defined through configuration dialogs. An installation might set up a group to refer to a physical site or an organizational or application entity.

group entry
A construct, created with the customization dialogs, used to represent and contain policy for a group.

group entry type
A collection of target systems defined through the customization dialog. An installation might set up a group to refer to a physical site or an organizational entity. Groups can, for example, be of type STANDARD or SYSPLEX.

H

Hardware Management Console (HMC)
A user interface through which data center personnel configure, control, monitor, and manage IBM Z hardware and software resources. The HMC communicates with each defined central processor complex (CPC) through the Support Element.

Hardware Management Console Application (HWMCA)
A direct-manipulation object-oriented graphical user interface that provides a single point of control and single system image for hardware elements. The HWMCA provides grouping support, aggregated and real-time system status using colors, consolidated hardware messages support, consolidated operating system messages support, consolidated service support, and hardware commands targeted at a single system, multiple systems, or a group of systems.

help panel
An online panel that tells you how to use a command or another aspect of a product.

hierarchy
In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

high-level language (HLL)
A programming language that provides some level of abstraction from assembler language and independence from a particular type of machine. For the NetView program, the high-level languages are PL/I and C.

HLL
See high-level language.

host (primary processor)
The processor that you enter a command at (also known as the *issuing processor*).

host system
In a coupled system or distributed system environment, the system on which the facilities for centralized automation run. SA z/OS publications refer to target systems or focal-point systems instead of hosts.

HWMCA
See Hardware Management Console Application.

Hypervisor
A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM, and PowerVM® Enterprise Edition.

I

IBM Secure Service Container (SSC)
IBM Z partitions, activated to run in SSC operating mode, provide the basic infrastructure runtime and deployment support for firmware or software based appliances, such as zAware or z/VSE® VNA.

IBM System z Application Assist Processor (zAAP)
A specialized processor that provides a Java execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

IBM System z Integrated Information Processor (zIIP)
See Integrated Information Processor (IIP).

IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z)
A purpose-built appliance that simplifies, helps secure, and optimizes XML and Web services processing.

IBM Workload Scheduler (IWS)
See ZWS.

IBM Z Workload Scheduler (ZWS)
The scheduler that plans, executes, and tracks jobs in z/OS environments. It's previously called IBM Workload Scheduler for z/OS (IWS), IBM Tivoli Workload Scheduler for z/OS (TWS), or OPC/A.

I/O resource number
Combination of channel path identifier (CHPID), device number, etc. See internal token.

images
A grouping of processors and I/O devices that you define. You can define a single-image mode that allows a multiprocessor system to function as one central processor image.

IMS
See Information Management System.

IMS/VS
See Information Management System/Virtual Storage.

inbound
In SA z/OS, messages sent to the focal-point system from the PC or target system.

inbound gateway operator
The automation operator that receives incoming messages, commands, and responses from the outbound gateway operator at the sending system. The inbound gateway operator handles communications with other systems using a gateway session.

Information Management System (IMS)
Any of several system environments available with a database manager and transaction processing that are capable of managing complex databases and terminal networks.

Information Management System/Virtual Storage (IMS/VS)
A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with Information Management System.

initial microprogram load
The action of loading microprograms into computer storage.

initial program load (IPL)
The initialization procedure that causes an operating system to commence operation.

The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction.

The process of loading system programs and preparing a system to run jobs.

initialize automation
SA z/OS-provided automation that issues the correct z/OS start command for each subsystem when SA z/OS is initialized. The automation ensures that subsystems are started in the order specified in the automation control files and that prerequisite applications are functional.

input/output configuration data set (IOCDS)
A configuration definition built by the I/O configuration program (IOCP) and stored on disk files associated with the processor controller.

input/output support processor (IOSP)
The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.

Integrated Information Processor (IIP)
A specialized processor that provides computing capacity for selected data and transaction processing workloads and for selected network encryption workloads.

Interactive System Productivity Facility (ISPF)
An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user. See also Time Sharing Option.

interested operator list
The list of operators who are to receive messages from a specific target system.

internal token
A *logical token* (LTOK); name by which the I/O resource or object is known; stored in IODF.

IOCDS
See input/output configuration data set.

IOSP
See input/output support processor..

IPL
See initial program load.

ISPF
See Interactive System Productivity Facility.

ISPF console
You log on to ISPF from this 3270-type console to use the runtime panels for SA z/OS customization panels.

issuing host
The base program that you enter a command for processing with. See primary host.

J

JCL
See job control language.

JES
See job entry subsystem.

JES2
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing. See also job entry subsystem and JES3

JES3
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them using a common job queue. See also job entry subsystem and JES2.

job
A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system.

An address space.

job control language (JCL)
A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES)
An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs. In SA z/OS publications, JES refers to JES2 or JES3, unless otherwise stated. See also JES2 and JES3.

K

Kanji
An ideographic character set used in Japanese. See also double-byte character set.

L

LAN
See local area network.

line mode
A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with full-screen mode.

link
In SNA, the combination of the link connection and the link stations joining network nodes; for example, a System/370 channel and its associated protocols, a serial-by-bit connection under the control of synchronous data link control (SDLC). See synchronous data link control.

In SA z/OS, link connection is the physical medium of transmission.

link-attached
Describes devices that are physically connected by a telecommunication line. Contrast with channel-attached.

Linux on z Systems
UNIX-like open source operating system conceived by Linus Torvalds and developed across the internet.

local
Pertaining to a device accessed directly without use of a telecommunication line. Synonymous with channel-attached.

local area network (LAN)
A network in which a set of devices is connected for communication. They can be connected to a larger network. See also token ring.

A network that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

logical partition (LP)
A subset of the processor hardware that is defined to support an operating system. See also logically partitioned mode.

logical token (LTOK)
Resource number of an object in the IODF.

logical unit (LU)
In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other LUs. See also physical unit and system services control point.

logical unit 6.2 (LU 6.2)
A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by:

- A peer relationship between session partners
- Efficient use of a session for multiple transactions
- A comprehensive end-to-end error processing
- A generic application program interface (API) consisting of structured verbs that are mapped to a product implementation

Synonym for advanced program-to-program communication.

logically partitioned (LPAR) mode
A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with basic mode.

LOGR
The sysplex logger.

LP
See logical partition.

LPAR
See logically partitioned mode.

LU
See logical unit.

LU 6.2
See logical unit 6.2.

LU 6.2 session
A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro. See logical unit 6.2.

LU-LU session
In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

M

MAT
Deprecated term for NetView automation table.

MCA
See Micro Channel architecture.

MCS
See multiple console support.

member
A specific function (one or more modules or routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

message automation table (MAT)
Deprecated term for NetView automation table.

message class
A number that SA z/OS associates with a message to control routing of the message. During automated operations, the classes associated with each message issued by SA z/OS are compared to the classes assigned to each notification operator. Any operator with a class matching one of the message's classes receives the message.

message forwarding
The SA z/OS process of sending messages generated at an SA z/OS target system to the SA z/OS focal-point system.

message group
Several messages that are displayed together as a unit.

message monitor task
A task that starts and is associated with a number of communications tasks. Message monitor tasks receive inbound messages from a communications task, determine the originating target system, and route the messages to the appropriate target control tasks.

message processing facility (MPF)
A z/OS table that screens all messages sent to the z/OS console. The MPF compares these messages with a customer-defined list of messages (based on this message list, messages are automated and/or suppressed from z/OS console display), and marks messages to automate or suppress. Messages are then broadcast on the subsystem interface (SSI).

message suppression
The ability to restrict the amount of message traffic displayed on the z/OS console.

Micro Channel architecture
The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

microprocessor
A processor implemented on one or a small number of chips.

migration
Installation of a new version or release of a program to replace an earlier version or release.

MP
Multiprocessor.

MPF
See message processing facility.

MPFLSTxx
The MPFLST member that is built by SA z/OS.

multi-MVS environment
physical processing system that is capable of operating more than one MVS image. See also MVS image.

multiple console support (MCS)
A feature of MVS that permits selective message routing to multiple consoles.

Multiple Virtual Storage (MVS)
An IBM operating system that accesses multiple address spaces in virtual storage. The predecessor of z/OS.

multiprocessor (MP)
A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application
An application program that has various functions distributed across z/OS images in a multisystem environment.

multisystem environment
An environment in which two or more systems reside on one or more processors. Or one or more processors can communicate with programs on the other systems.

MVS
See Multiple Virtual Storage.

MVS image
A single occurrence of the MVS operating system that has the ability to process work. See also multi-MVS environment and single-MVS environment.

MVS/ESA
Multiple Virtual Storage/Enterprise Systems Architecture. See z/OS.

MVS/JES2
Multiple Virtual Storage/Job Entry System 2. A z/OS subsystem that receives jobs into the system, converts them to an internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

N

NAU
See network addressable unit.

See network accessible unit.

NCCF
See Network Communications Control Facility..

NCP
See network control program (general term).

See Network Control Program (an IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with ACF/NCP.

NCP/token ring interconnection
A function used by ACF/NCP to support token ring-attached SNA devices. NTRI also provides translation from token ring-attached SNA devices (PUs) to switched (dial-up) devices.

NetView
An IBM licensed program used to monitor a network, manage it, and diagnose network problems. NetView consists of a command facility that includes a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the session monitor, hardware monitor, and terminal access facility (TAF) network management applications are built.

NetView (NCCF) console
A 3270-type console for NetView commands and runtime panels for system operations and processor operations.

NetView automation procedures
A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program.

NetView automation table (AT)
A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA z/OS entries in the NetView automation table trigger an SA z/OS response to target system conditions. Formerly known as the message automation table (MAT).

NetView command list language
An interpretive language unique to NetView that is used to write command lists.

NetView hardware monitor
The component of NetView that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. Formerly called *network problem determination application*.

NetView log
The log that NetView records events relating to NetView and SA z/OS activities in.

NetView message table
See NetView automation table.

NetView paths via logical unit (LU 6.2)
A type of network-accessible port (VTAM connection) that enables end users to gain access to SNA network resources and communicate with each other. LU 6.2 permits communication between processor operations and the workstation. See logical unit 6.2.

NetView-NetView task (NNT)
The task that a cross-domain NetView operator session runs under. Each NetView program must have a NetView-NetView task to establish one NNT session. See also operator station task.

NetView-NetView task session
A session between two NetView programs that runs under a NetView-NetView task. In SA z/OS, NetView-NetView task sessions are used for communication between focal point and remote systems.

network
An interconnected group of nodes.

In data processing, a user application network. See SNA network.

network accessible unit (NAU)
In SNA networking, any device on the network that has a network address, including a logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with network addressable unit.

network addressable unit (NAU)
Synonym for network accessible unit.

Network Communications Control Facility (NCCF)
The operations control facility for the network. NCCF consists of a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the network management applications NLDM are built. NCCF is a precursor to the NetView command facility.

Network Control Program (NCP)
An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

network control program (NCP)
A program that controls the operation of a communication controller.

A program used for requests and responses exchanged between physical units in a network for data flow control.

Networking NetView
In SA z/OS the NetView that performs network management functions, such as managing the configuration of a network. In SA z/OS it is common to also route alerts to the Networking NetView.

NIP
See nucleus initialization program.

NNT
See NetView-NetView task.

notification message
An SA z/OS message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialogs.

notification operator
A NetView console operator who is authorized to receive SA z/OS notification messages. Authorization is made through the customization dialogs.

NTRI
See NCP/token ring interconnection.

nucleus initialization program (NIP)
The program that initializes the resident control program; it allows the operator to request last-minute changes to certain options specified during system generation.

O

objective value
An average Workflow or Using value that SA z/OS can calculate for applications from past service data. SA z/OS uses the objective value to calculate warning and alert thresholds when none are explicitly defined.

OCA
In SA z/OS, operator console A, the active operator console for a target system. Contrast with OCB.

OCB
In SA z/OS, operator console B, the backup operator console for a target system. Contrast with OCA.

OPC/A
See Operations Planning and Control/Advanced.

OPC/ESA
See Operations Planning and Control/Enterprise Systems Architecture.

operating system (OS)
Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

operations
The real-time control of a hardware device or software function.

Operations Planning and Control/Advanced (OPC/A)
A set of IBM licensed programs that automate, plan, and control batch workload. OPC/A analyzes system and workload status and submits jobs accordingly.

Operations Planning and Control/Enterprise Systems Architecture (OPC/ESA)
A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

operator
A person who keeps a system running.

A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS.

A person who operates a device.

In a language statement, the lexical entity that indicates the action to be performed on operands.

operator console
A functional unit containing devices that are used for communications between a computer operator and a computer. (T)

A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and to monitor system operation.

In SA z/OS, a console that displays output from and sends input to the operating system (z/OS, LINUX, VM, VSE). Also called *operating system console*. In the SA z/OS operator commands and configuration dialogs, OC is used to designate a target system operator console.

operator station task (OST)
The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

operator view
A set of group, system, and resource definitions that are associated together for monitoring purposes. An operator view appears as a graphic display in the graphical interface showing the status of the defined groups, systems, and resources.

OperatorView entry
A construct, created with the customization dialogs, used to represent and contain policy for an operator view.

optimizer
A special-purpose hardware component or appliance that can perform a limited set of specific functions with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a stand-alone unit.

OS
See operating system.

OST
See operator station task.

outbound
In SA z/OS, messages or commands from the focal-point system to the target system.

outbound gateway operator
The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

P

page
The portion of a panel that is shown on a display surface at one time.

To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel
A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics.

By using SA z/OS panels you can see status, type commands on a command line using a keyboard, configure your system, and passthru to other consoles. See also help panel.

In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with screen.

parameter
A variable that is given a constant value for a specified application and that may represent an application, for example.

An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted.

Data passed to a program or procedure by a user or another program, specifically as an operand in a language statement, as an item in a menu, or as a shared data structure.

partition
A fixed-size division of storage.

In VSE, a division of the virtual address area that is available for program processing.

On an IBM Personal Computer fixed disk, one of four possible storage areas of variable size; one can be accessed by DOS, and each of the others may be assigned to another operating system.

partitionable CPC
A CPC that can be divided into 2 independent CPCs. See also physical partition, single-image mode, MP, and side.

partitioned data set (PDS)
A data set in direct access storage that is divided into partitions, called *members*, each of which can contain a program, part of a program, or data.

passive monitoring
In SA z/OS, the receiving of unsolicited messages from z/OS systems and their resources. These messages can prompt updates to resource status displays. See also active monitoring

PCE
A processor controller. Also known as the support processor or service processor in some processor families.

PDB
See policy database.

PDS
See partitioned data set.

physical partition
Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physical unit (PU)
In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

physically partitioned (PP) configuration
A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, utilities, and maintenance boundaries. Contrast with single-image mode.

PLEXID group
PLEXID group or "extended XCF communication group" is a term used in conjunction with a sysplex. The PLEXID group includes System Automation Agents for a subset of a sysplex or for the entire sysplex. It is used to provide XCF communication beyond the SAplex boundaries. For a detailed description, refer to "Defining the Extended XCF Communication Group" in *IBM Z System Automation Planning and Installation*.

POI
See program operator interface.

policy
The automation and monitoring specifications for an SA z/OS enterprise. See *IBM Z System Automation Defining Automation Policy*.

policy database
The automation definitions (automation policy) that the automation administrator specifies using the customization dialog is stored in the policy database. Also known as the PDB. See also automation policy.

POR
See power-on reset.

port
System hardware that the I/O devices are attached to.

An access point (for example, a logical unit) for data entry or exit.

A functional unit of a node that data can enter or leave a data network through.

In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices.

power-on reset (POR)
A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system. See initial microprogram load.

PP
See physical partition.

PPI
See program to program interface.

PPT
See primary POI task.

PR/SM
See Processor Resource/Systems Manager.

primary host
The base program that you enter a command for processing at.

primary POI task (PPT)
The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

primary system
A system is a primary system for an application if the application is normally meant to be running there. SA z/OS starts the application on all the primary systems defined for it.

problem determination
The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environment failure such as a power loss, or user error.

processor
A device for processing data from programmed instructions. It may be part of another unit.

In a computer, the part that interprets and executes instructions. Two typical components of a processor are a control unit and an arithmetic logic unit.

processor controller
Hardware that provides support and diagnostic functions for the central processors.

processor operations
The part of SA z/OS that monitors and controls processor (hardware) operations. Processor operations provides a connection from a focal-point system to a target system. Through NetView on the focal-point system, processor operations automates operator and system consoles for monitoring and recovering target systems. Also known as ProcOps.

Processor Resource/Systems Manager (PR/SM)
The feature that allows the processor to use several operating system images simultaneously and provides logical partitioning capability. See also logically partitioned mode.

ProcOps
See processor operations.

ProcOps Service Machine (PSM)
The PSM is a CMS user on a VM host system. It runs a CMS multitasking application that serves as "virtual hardware" for ProcOps. ProOps communicates via the PSM with the VM guest systems that are defined as target systems within ProcOps.

product automation
Automation integrated into the base of SA z/OS for the products CICS, Db2, IMS, IBM Workload Scheduler (formerly called *features*).

program operator interface (POI)
A NetView facility for receiving VTAM messages.

program to program interface (PPI)
A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

protocol
In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

proxy resource
A resource defined like an entry type APL representing a processor operations target system.

PSM
See ProcOps Service Machine.

PU
See physical unit.

R

RACF
See Resource Access Control Facility.

remote system
A system that receives resource status information from an SA z/OS focal-point system. An SA z/OS remote system is defined as part of the same SA z/OS enterprise as the SA z/OS focal-point system to which it is related.

requester
A workstation from that user can log on to a domain from, that is, to the servers belonging to the domain, and use network resources. Users can access the shared resources and use the processing capability of the servers, thus reducing hardware investment.

resource
Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.

In NetView, any hardware or software that provides function to the network.

In SA z/OS, any z/OS application, z/OS component, job, device, or target system capable of being monitored or automated through SA z/OS.

Resource Access Control Facility (RACF)
A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.

resource group
A physically partitionable portion of a processor. Also known as a *side*.

Resource Measurement Facility (RMF)
A feature of z/OS that measures selected areas of system activity and presents the data collected in the format of printed reports, System Management Facility (SMF) records, or display reports.

restart automation
Automation provided by SA z/OS that monitors subsystems to ensure that they are running. If a subsystem fails, SA z/OS attempts to restart it according to the policy in the automation configuration file.

Restructured Extended Executor (REXX)
A general-purpose, high-level, programming language, particularly suitable for EXEC procedures or programs for personal computing, used to write command lists.

return code
A code returned from a program used to influence the issuing of subsequent instructions.

REXX
See Restructured Extended Executor.

REXX procedure
A command list written with the Restructured Extended Executor (REXX), which is an interpretive language.

RMF
See Resource Measurement Facility.

S

SAF
See Security Authorization Facility.

SA IOM
See System Automation for Integrated Operations Management.

SAplex
SAplex or "SA z/OS Subplex" is a term used in conjuction with a sysplex. In fact, a SAplex is a subset of a sysplex. However, it can also be a sysplex. For a detailed description, refer to "Using SA z/OS Subplexes" in *IBM Z System Automation Planning and Installation*.

SA z/OS
See “IBM Z System Automation” on page 267.

SA z/OS customization dialogs
An ISPF application through which the SA z/OS policy administrator defines policy for individual z/OS systems and builds automation control data.

SA z/OS customization focal point system
See focal point system.

SA z/OS data model
The set of objects, classes and entity relationships necessary to support the function of SA z/OS and the NetView automation platform.

SA z/OS enterprise
The group of systems and resources defined in the customization dialogs under one enterprise name. An SA z/OS enterprise consists of connected z/OS systems running SA z/OS.

SA z/OS focal point system
See focal point system.

SA z/OS policy
The description of the systems and resources that make up an SA z/OS enterprise, together with their monitoring and automation definitions.

SA z/OS policy administrator
The member of the operations staff who is responsible for defining SA z/OS policy.

SA z/OS SDF focal point system
See focal point system.

SCA
In SA z/OS, system console A, the active system console for a target hardware. Contrast with SCB.

SCB
In SA z/OS, system console B, the backup system console for a target hardware. Contrast with SCA.

screen
Deprecated term for panel.

screen handler
In SA z/OS, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

SDF
See status display facility.

SDLC
See synchronous data link control.

SDSF
See System Display and Search Facility.

secondary system
A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable. SA z/OS does not start the application on its secondary systems.

Security Authorization Facility (SAF)
An MVS interface with which programs can communicate with an external security manager, such as RACF.

server
A server is a workstation that shares resources, which include directories, printers, serial devices, and computing powers.

service language command (SLC)
The line-oriented command language of processor controllers or service processors.

service period
Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

service processor (SVP)
The name given to a processor controller on smaller System/370 processors.

service threshold
An SA z/OS policy setting that determines when to notify the operator of deteriorating service for a resource. See also alert threshold and warning threshold.

session
In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

session monitor
The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDM.

shutdown automation
SA z/OS-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

side
A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

Simple Network Management Protocol (SNMP)
A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB).

single image
A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.

single-MVS environment
An environment that supports one MVS image. See also MVS image.

single-image (SI) mode
A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with physically partitioned (PP) configuration.

SLC
See service language command.

SMP/E
See System Modification Program/Extended.

SNA
See Systems Network Architecture.

SNA network
In SNA, the part of a user-application network that conforms to the formats and protocols of systems network architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNMP
See Simple Network Management Protocol.

solicited message
An SA z/OS message that directly responds to a command. Contrast with unsolicited message.

SSCP
See system services control point.

SSI
See subsystem interface.

start automation
Automation provided by SA z/OS that manages and completes the startup process for subsystems. During this process, SA z/OS replies to prompts for additional information, ensures that the startup process completes within specified time limits, notifies the operator of problems, if necessary, and brings subsystems to an UP (or ready) state.

startup
The point in time that a subsystem or application is started.

status
The measure of the condition or availability of the resource.

status display facility (SDF)
The system operations part of SA z/OS that displays status of resources such as applications, gateways, and write-to-operator messages (WTORs) on dynamic color-coded panels. SDF shows spool usage problems and resource data from multiple systems.

steady state automation
The routine monitoring, both for presence and performance, of subsystems, applications, volumes and systems. Steady state automation may respond to messages, performance exceptions and discrepancies between its model of the system and reality.

structure
A construct used by z/OS to map and manage storage on a coupling facility.

subgroup
A named set of systems. A subgroup is part of an SA z/OS enterprise definition and is used for monitoring purposes.

SubGroup entry
A construct, created with the customization dialogs, used to represent and contain policy for a subgroup.

subplex
See SAplex.

subsystem
A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

subsystem interface (SSI)
The z/OS interface over which all messages sent to the z/OS console are broadcast.

support element (SE)
A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

support processor
Another name given to a processor controller on smaller System/370 processors. See service processor.

SVP
See service processor.

symbolic destination name (SDN)
Used locally at the workstation to relate to the VTAM application name.

synchronous data link control (SDLC)
A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

SYSINFO Report
An RMF report that presents an overview of the system, its workload, and the total number of jobs using resources or delayed for resources.

SysOps
See system operations.

sysplex
A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and timers) and software services (couple data sets).

In a sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors, sysplex timers, coupling facilities, and couple data sets (which contains policy and states for automation).

A Parallel Sysplex is a sysplex that includes a coupling facility.

sysplex application group
A sysplex application group is a grouping of applications that can run on any system in a sysplex.

sysplex couple data set
A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also couple data set.

Sysplex Timer
An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

system
In SA z/OS, system means a focal point system (z/OS) or a target system (MVS, VM, VSE, LINUX, or CF).

System Automation for Integrated Operations Management
An outboard automation solution for secure remote access to mainframe/distributed systems. Tivoli System Automation for Integrated Operations Management, previously Tivoli AF/REMOTE, allows users to manage mainframe and distributed systems from any location.

The full name for SA IOM.

IBM Z System Automation
The full name for SA z/OS.

system console
A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system.

A logical device used for the operation and control of hardware functions (for example, IPL, alter/display, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor.

In SA z/OS, the hardware system console for processor controllers or service processors of processors connected using SA z/OS. In the SA z/OS operator commands and configuration dialogs, SC is used to designate the system console for a target hardware processor.

System Display and Search Facility (SDSF)
An IBM licensed program that provides information about jobs, queues, and printers running under JES2 on a series of panels. Under SA z/OS you can select SDSF from a pull-down menu to see the resources' status, view the z/OS system log, see WTOR messages, and see active jobs on the system.

System entry
A construct, created with the customization dialogs, used to represent and contain policy for a system.

System Modification Program/Extended (SMP/E)
An IBM licensed program that facilitates the process of installing and servicing an z/OS system.

system operations
The part of SA z/OS that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, ACF/VTAM, CICS, IMS, and OPC. Also known as SysOps.

system services control point (SSCP)
In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

System/390 microprocessor cluster
A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

Systems Network Architecture (SNA)
The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

T

TAF
See terminal access facility.

target
A processor or system monitored and controlled by a focal-point system.

target control task
In SA z/OS, target control tasks process commands and send data to target systems and workstations through communications tasks. A target control task (a NetView autotask) is assigned to a target system when the target system is initialized.

target hardware
In SA z/OS, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with target system.

target system
In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system.

In SA z/OS, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

task
A basic unit of work to be accomplished by a computer.

In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA z/OS tasks are NetView tasks. See also message monitor task, and target control task.

telecommunication line
Any physical medium, such as a wire or microwave beam, that is used to transmit data.

terminal access facility (TAF)
A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA z/OS customization panels so you don't have to set them up each time you want to use them.

In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

terminal emulation
The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

threshold
A value that determines the point at which SA z/OS automation performs a predefined action. See alert threshold, warning threshold, and error threshold.

time of day (TOD)
Typically refers to the time-of-day clock.

Time Sharing Option (TSO)
An optional configuration of the operating system that provides conversational time sharing from remote stations. It is an interactive service on z/OS, MVS/ESA, and MVS/XA.

Time-Sharing Option/Extended (TSO/E)
An option of z/OS that provides conversational timesharing from remote terminals. TSO/E allows a wide variety of users to perform many different kinds of tasks. It can handle short-running applications that use fewer sources as well as long-running applications that require large amounts of resources.

timers
A NetView instruction that issues a command or command processor (list of commands) at a specified time or time interval.

TOD
Time of day.

token ring
A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network product.

TP
See transaction program.

transaction program
In the VTAM program, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the

VTAM application program interface (API). In that situation, the transaction program would request services from the applications program using protocols defined by that application program. The application program, in turn, could request services from the VTAM program by issuing the APPCCMD macro instruction.

transitional automation
The actions involved in starting and stopping subsystems and applications that have been defined to SA z/OS. This can include issuing commands and responding to messages.

translating host
Role played by a host that turns a resource number into a token during a unification process.

trigger
Triggers, in combination with events and service periods, are used to control the starting and stopping of applications in a single system or a parallel sysplex.

TSO
See Time Sharing Option.

TSO console
From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for SA z/OS customization panels.

TSO/E
See Time-Sharing Option/Extended.

TWS
See ZWS.

U

unsolicited message
An SA z/OS message that is not a direct response to a command.

uniform resource identifier (URI)
A uniform resource identifier is a string of characters used to identify a name of a web resource. Such identification enables interaction with representations of the web resource over the internet, using specific protocols.

user task
An application of the NetView program defined in a NetView TASK definition statement.

Using
An RMF Monitor III definition. Jobs getting service from hardware resources (processors or devices) are **using** these resources. The use of a resource by an address space can vary from 0% to 100% where 0% indicates no use during a Range period, and 100% indicates that the address space was found using the resource in every sample during that period.

V

view
In the NetView Graphic Monitor Facility, a graphical picture of a network or part of a network. A view consists of nodes connected by links and may also include text and background lines. A view can be displayed, edited, and monitored for status information about network resources.

Virtual Server
A logical construct that appears to comprise processor, memory, and I/O resources conforming to a particular architecture. A virtual server can support an operating system, associated middleware, and applications. A hypervisor creates and manages virtual servers.

Virtual Server Collection
A set of virtual servers that supports a workload. This set is not necessarily static. The constituents of the collection at any given point are determined by virtual servers involved in supporting the workload at that time.

virtual Server Image
A package containing metadata that describes the system requirements, virtual storage drives, and any goals and constraints for the virtual machine {for example, isolation and availability). The Open

Virtual Machine Format (OVF) is a Distributed Management Task Force (DMTF) standard that describes a packaging format for virtual server images.

Virtual Server Image Capture
The ability to store metadata and disk images of an existing virtual server. The metadata describes the virtual server storage, network needs, goals and constraints. The captured information is stored as a virtual server image that can be referenced and used to create and deploy other similar images.

Virtual Server Image Clone
The ability to create an identical copy (clone) of a virtual server image that can be used to create a new similar virtual server.

Virtual Storage Extended (VSE)
A system that consists of a basic operating system (VSE/Advanced Functions), and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware that it controls form a complete computing system. Its current version is called VSE/ESA.

Virtual Telecommunications Access Method (VTAM)
An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with ACF/VTAM.

VM Second Level Systems Support
With this function, Processor Operations is able to control VM second level systems (VM guest systems) in the same way that it controls systems running on real hardware.

VM/ESA
Virtual Machine/Enterprise Systems Architecture. Its current version is called z/VM.

volume
A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

VSE
See Virtual Storage Extended.

VTAM
See Virtual Telecommunications Access Method.

W

warning threshold
An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the warning color. See alert threshold.

workstation
In SA z/OS workstation means the *graphic workstation* that an operator uses for day-to-day operations.

write-to-operator (WTO)
A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

write-to-operator-with-reply (WTOR)
A request to send a message to an operator at the z/OS operator console that requires a response from the operator. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

WTO
See write-to-operator.

WTOR
See write-to-operator-with-reply.

WWV
The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

X

XCF
See cross-system coupling facility.

XCF couple data set
The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See also sysplex couple data set.

XCF group
A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

XRF
See extended recovery facility.

Z

z/OS
An IBM mainframe operating system that uses 64-bit real storage. See also Base Control Program.

z/OS component
A part of z/OS that performs a specific z/OS function. In SA z/OS, component refers to entities that are managed by SA z/OS automation.

z/OS subsystem
Software products that augment the z/OS operating system. JES and TSO/E are examples of z/OS subsystems. SA z/OS includes automation for some z/OS subsystems.

z/OS system
A z/OS image together with its associated hardware, which collectively are often referred to simply as a system, or z/OS system.

zAAP
See IBM System z Application Assist Processor (zAAP).

zCPC
The physical collection of main storage, central processors, timers, and channels within a zEnterprise mainframe. See also central processor complex.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Z

SC34-2718-04