System Automation for OS/390

Planning and Installation

Version 2 Release 2

SC33-7038-05

System Automation for OS/390

Planning and Installation

Version 2 Release 2

SC33-7038-05

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xi.

Ninth Edition (July 2003)

This edition applies to System Automation for OS/390 (Program Number 5645-006) Version 2, Release 2, an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

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

IBM welcomes your comments. A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:
IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

If you prefer to send comments electronically, use one of the following methods:
FAX (Germany): 07031 + 16-3456
FAX (Other Countries): (+49)+7031-16-3456
Internet: s390id@de.ibm.com

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

Contents

Chapter 7. Naming Conventions 55

Part 2. Installation 61

Chapter 8. Installing SA OS/390 on Host Systems. 63

Chapter 9. Installing SA OS/390 on Workstations 129

Part 3. Appendixes 143

Appendix A. Security and Authorization 145

Appendix B. Syntax for INGTOPOF File 167

Appendix C. Miscellaneous Information 177

Appendix D. Processor Operations Sample 181

Appendix E. Migration Information 183

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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

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

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

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

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

Web Site Disclaimer

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

Programming Interface Information

This publication documents information that is NOT intended to be used as a Programming Interface of System Automation for OS/390.

Trademarks

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

ACF/VTAM
APPN
CICS
DB2
DFSMS/MVS
ESCON
FICON
IBM
IMS
Library Reader
Micro Channel
MQSeries
Multiprise
MVS
MVS/ESA
Netfinity
Netfinity Manager
NetView
OpenEdition
OS/2
OS/390
Parallel Sysplex
RACF
RMF
Sysplex Timer
System/390
Tivoli
Tivoli Enterprise Console
VM/ESA
VSE/ESA
VTAM
z/OS
zSeries

The following terms are trademarks of other companies:

HeathKit
Heath Co., St. Joseph, MO

Spectracom
Spectracom Corp., Rochester, NY

Windows
Microsoft Corporation

Windows NT
Microsoft Corporation

Java Sun Microsystems, Inc.

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

LINUX
LINUX is a registered trademark of Linus Torvalds and others.

About This Book

This book describes **System Automation for OS/390** (SA OS/390) from a planning point of view, and how to install the product.

| **Note:** Those sections of this book that are marked with a revision bar, ′|′, describe
| enhanced automation functions that are delivered with APAR OA02321
| (07/2003). This functionality is only available if you have installed
| APAR OA02321.

Who Should Use This Book

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

Notes on Terminology

MVS:

References in this book to ″MVS″ refer either to the MVS/ESA™ product or to the MVS™ element of OS/390.

NetView:

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

Where to Find More Information

The System Automation for OS/390 Library

The following table shows the information units in the System Automation for OS/390 library:

Table 1. System Automation for OS/390 Library

Title	Order Number
System Automation for OS/390 Planning and Installation	SC33-7038
System Automation for OS/390 Customizing and Programming	SC33-7035
System Automation for OS/390 Defining Automation Policy	SC33-7039
System Automation for OS/390 User's Guide	SC33-7040
System Automation for OS/390 Messages and Codes	SC33-7041
System Automation for OS/390 Operator's Commands	SC33-7042
System Automation for OS/390 Programmer's Reference	SC33-7043
System Automation for OS/390 CICS Automation Programmer's Reference and Operator's Guide	SC33-7044
System Automation for OS/390 IMS Automation Programmer's Reference and Operator's Guide	SC33-7045
System Automation for OS/390 OPC Automation Programmer's Reference and Operator's Guide	SC23-7046
System Automation for OS/390 Licensed Program Specifications	SC33-7037

The System Automation for OS/390 books (except Licensed Program Specifications) are also available on CD-ROM as part of the following collection kit:

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

SA OS/390 Homepage

For the latest news on SA OS/390, visit the SA OS/390 homepage at http://www.ibm.com/servers/eserver/zseries/software/sa

Related Product Information

The following table shows the books in the related product libraries that you may find useful for support of the SA OS/390 base program.

Table 2. Related Products Books

Title	Order Number
ISPF User's Guide	SC34-4484
ISPF Dialog Management Guide and Reference	SC34-4266
MVS/ESA MVS Configuration Program Guide and Reference	GC28-1817
MVS/ESA Planning: Dynamic I/O Configuration	GC28-1674
MVS/ESA Support for the Enterprise Systems Connection	GC28-1140
MVS/ESA Planning: APPC Management	GC28-1110
MVS/ESA Application Development Macro Reference	GC28-1822
OS/390: MVS System Commands	GC28-1781
MVS/ESA SPL Application Development Macro Reference	GC28-1857
OS/390 Hardware Configuration Definition: User's Guide	SC28-1848
OS/390 Information Roadmap	GC28-1727
OS/390 Information Transformation	GC28-1985
OS/390 Introduction and Release Guide	GC28-1725

Table 2. Related Products Books (continued)

Title	Order Number
OS/390 JES Commands Summary	GX22-0041
OS/390 Licensed Program Specifications	GC28-1728
OS/390 Printing Softcopy Books	S544-5354
OS/390 Starting Up a Sysplex	GC28-1779
OS/390 Up and Running!	GC28-1726
Planning for the 9032 Model 3 and 9033 Enterprise Systems Connection Director	SA26-6100
Resource Access Control Facility (RACF) Command Language Reference	SC28-0733
S/390 MVS Sysplex Overview – An Introduction to Data Sharing and Parallelism	GC23-1208
S/390 MVS Sysplex Systems Management	GC23-1209
S/390 Sysplex Hardware and Software Migration	GC23-1210
S/390 MVS Sysplex Application Migration	GC23-1211
S/390 Managing Your Processors	GC38-0452
Tivoli/Enterprise Console User's Guide Volume I	GC31-8334
Tivoli/Enterprise Console User's Guide Volume II	GC31-8335
Tivoli/Enterprise Console Event Integration Facility Guide	GC31-8337
Tivoli NetView for OS/390 Administration Reference	SC31-8222
Tivoli NetView for OS/390 Application Programming Guide	SC31-8223
Tivoli NetView for OS/390 APPN Topology and Accounting Agent	SC31-8224
Tivoli NetView for OS/390 Automation Guide	SC31-8225
Tivoli NetView for OS/390 AON Customization Guide	SC31-8662
Tivoli NetView for OS/390 AON User's Guide	GC31-8661
Tivoli NetView for OS/390 Bridge Implementation	SC31-8238
Tivoli NetView for OS/390 Command Reference Vol. 1	SC31-8227
Tivoli NetView for OS/390 Command Reference Vol. 2	SC31-8735
Tivoli NetView for OS/390 Customization Guide	SC31-8228
Tivoli NetView for OS/390 Customization: Using Assembler	SC31-8229
Tivoli NetView for OS/390 Customization: Using Pipes	SC31-8248
Tivoli NetView for OS/390 Customization: Using PL/I and C	SC31-8230
Tivoli NetView for OS/390 Customization: Using REXX and CLIST Language	SC31-8231
Tivoli NetView for OS/390 Data Mode Reference	SC31-8232
Tivoli NetView for OS/390 Installation: Getting Started	SC31-8767
Tivoli NetView for OS/390 Installation: Migration Guide	SC31-8768
Tivoli NetView for OS/390 Installation: Configuring Graphical Components	SC31-8770
Tivoli NetView for OS/390 Installation: Configuring Additional Components	SC31-8769
Tivoli NetView for OS/390 Messages and Codes	SC31-8237
Tivoli NetView for OS/390 MultiSystem Manager User's Guide	SC31-8607

Table 2. Related Products Books (continued)

Title	Order Number
Tivoli NetView for OS/390 NetView Management Console User's Guide	GC31-8665
Tivoli NetView for OS/390 User's Guide	SC31-8241
Tivoli NetView for OS/390 RODM and GMFHS Programming Guide	SC31-8233
Tivoli NetView for OS/390 Security Reference	SC31-8606
Tivoli NetView for OS/390 SNA Topology Manager and APPN Accounting Manager Implementation Guide	SC31-8239
Tivoli Management Platform Reference Guide	GC31-8324
TSO/E REXX/MVS User's Guide	SC28-1882
TSO/E REXX/MVS Reference	SC28-1883
VM/XA SP GCS Command and Macro Reference	SC23-0433
VSE/SP Unattended Node Support	SC33-6412
VTAM Messages and Codes	SC31-6493
VTAM Network Implementation Guide	SC31-6404
VTAM Network Implementation Guide	SC31-6434

Using LookAt to look up message explanations

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

You can access LookAt from the Internet at: http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/ or from anywhere in z/OS or z/OS.e where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS).

The LookAt Web site also features a mobile edition of LookAt for devices such as Pocket PCs, Palm OS, or Linux-based handhelds. So, if you have a handheld device with wireless access and an Internet browser, you can now access LookAt message information from almost anywhere.

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the LookAt Web site's **Download** link.

Accessing z/OS™ licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

`http://www.ibm.com/servers/resourcelink`

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code. [1]

[1] z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

To obtain your IBM Resource Link user ID and password, log on to:

`http://www.ibm.com/servers/resourcelink`

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

Part 1. Planning

Chapter 1. SA OS/390 Prerequisites and Supported Equipment

SA OS/390 Components

Please note that SA OS/390 consists of the following three components:

- system operations, sometimes also abbreviated as *SysOps*
- processor operations, sometimes also abbreviated as *ProcOps*
- I/O operations, sometimes also abbreviated as *I/O Ops*

Refer to “Component Description” on page 15 for details.

SA OS/390 also provides special automation facilities for the following products:

- CICS®
- DB2™
- IMS®
- OPC

Hardware Requirements

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

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

SA OS/390 Processor Operations

The processor operations base program can run on any processor supported by NetView 1.4.

SA OS/390 System Operations

The system operations base program can run on any processor supported by NetView 1.4 and OS/390 Version 2 Release 10.

SA OS/390 I/O Operations

The I/O operations base program can run on any processor supported by OS/390 Version 2 Release 10.

Note: If an ESCON channel is not installed and defined, I/O operations recover from an ABEND 0C1, issue message IHVD014E, and terminate startup.

Workstation Components

The NMC exploitation used by SA OS/390 can run on all software supported by Tivoli NetView for OS/390 Release 4.

Functional Prerequisites

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

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

The following processor hardware can be controlled as a target with the BCP Internal Interface of the above listed processors, but cannot use the SA OS/390 BCP Internal Interface to control itself or other processors:

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

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

```
2064 (z900)  Driver 3C: J10638.039
                        J10638.064
                        J10638.072
             or
             Driver 3G: J11213.026

2066 (z800)  Driver 3G: J11213.026

9672 (G5/G6) Driver 26: F99918.102
                        F99918.106
                        F99918.115
                        F99918.119

9672 (G3/G4) Driver A2: F10980.081
```

Software Requirements

This section describes the environment of the target system required to install and use SA OS/390.

Notes:

1. For I/O operations only: To properly invoke the Japanese language version of this product, a Japanese language version host must be connected to a Japanese language workstation. If an English language workstation is connected to a Japanese language host, some messages may be unreadable.
2. Check with IBM Service for required product service levels in addition to the base product releases. Certain service levels may be required for particular product functions.
3. SA OS/390 processor operations is enabled on a focal-point system, from which it monitors and controls SA OS/390 processor operations target systems. The SA OS/390 processor operations target system may also have SA OS/390 installed for its system operations and I/O operations but the processor operations will not be enabled. This section does not describe the SA OS/390 Processor Operations target system.

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

Mandatory Prerequisites

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

Table 3. Mandatory Prerequisites

Product Name and Minimum VRM/Service Level
OS/390 Version 2 Release 10 or higher, including APAR OW48709
Tivoli NetView for OS/390 Release 4

Functional Prerequisites

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

Table 4. Functional Prerequisites

Product Name and Minimum VRM/Service Level	Function
OS/390 base elements or optional features:	
SecureWay Security Server for OS/390 (including RACF and DCE Security Server components), optional feature in OS/390 V2R10	For sysplex-based authorization and RACF-based NetView authorization.
Other program products:	
MQSeries Version 5 Release 2	For sysplex automation and using MQSeries as a communication vehicle.
HTML Browser	For customization reports. To view the HTML file by an Internet Browser, either IE 5.50 or above, or Netscape 4.72 or above.
Workstation Prerequisites:	
Tivoli NetView for OS/390 Release 4 MultiSystem Manager	For SA OS/390 topology manager functions
NetView Management Console Server and Client	For the NMC workstation
NetView 3270 Management Console	For the NMC workstation
APARs that must be installed:	
Service Processor Interface APAR OW51923	For the SA OS/390 BCP Internal Interface function used by the INGPLEX command and the IXC102A message automation without SA OS/390 processor operations. This APAR allows multiple applications (especially HCD and SA OS/390) to use the BCP Internal Interface simultaneously. SA OS/390 uses this interface to activate, inactivate, and query coupling facilities as well as to send hardware commands to the LPAR of the system which is being partitioned out of the sysplex.

Table 4. Functional Prerequisites (continued)

Product Name and Minimum VRM/Service Level	Function
XCF APARs OW52369 and OW53637	For the coupling facility drain and enable functions (INGCF DRAIN/ENABLE). With z/OS 1.2, XCF introduced a different behavior from previous functionality of how users are informed when a rebuild duplex process has completed. Only users starting the duplex process are informed when this process has been stopped. Therefore, all automation functions doing rebuilds will time out for structures whose duplex process has been started by MVS or by another operator. APAR OW52369 resolves the situation by informing the user who started the duplex process as well as the user who stops it. This new behavior also affects messages issued on behalf of the SETXCF ALTER command. When the automation rebuilds a structure, it also checks its initial and actual size. If the initial size is less than the actual size, the automation tries to change the initial size to the actual size. Because the expected message is no longer issued as a command response, the automation times out when waiting for it and issues an appropriate message. Even if automation continues the message could be misleading. APAR OW53637 resolves the problem.
GRS APARs OW53529 and OW56587	For the resolving of long running ENQs function. The automation of eliminating long-running ENQs is only available when these APARs are installed. After installing APAR OW56587, you can activate this automation by issuing the ACF COLD command from the NCCF screen of the NVSS where you want to run this sysplex-wide automation. This command will schedule the automation function on the appropriate autotask when APAR OW56587 has been applied. Recycling the NVSS address space has the same result.
Hardware Prerequisites:[1]	
Required Support Element LIC Levels	For current information about the required LIC levels for the following servers, refer to the PSP bucket. • zSeries • CMOS-S/390 G5, G6 • CMOS-S/390 G3, G4
Required Hardware Management Console LIC Levels	For zSeries processors, the following LIC levels are required: • Driver 3g, J11213.107 • Driver 3c, J10638.116 For CMOS-S/390 G5 and G6 processors the following LIC level is required: • Driver 26.F99918.140 These MCL levels are required for all HMCs that serve as Master HMCs and have the LIC change console service enabled. Note that at least one HMC in your processor LAN configuration must have this service enabled in order to provide cross-CPC communication over the BCP Internal Interface.

Table 4. Functional Prerequisites (continued)

Product Name and Minimum VRM/Service Level	Function
Notes: 1. A number of hardware commands are not supported when running on a z/OS image which runs under z/VM. Refer to*System Automation for OS/390 Customizing and Programming* for information about which particular functions are affected.	

Supported Hardware

SA OS/390 processor operations supports monitoring and control functions for any of the following processors:

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

SA OS/390 processor operations also supports logical partitioning of any of those processors.

SA OS/390 provides a wide range of I/O configuration information and control functions for various types of hardware other than processors, though it does not require any of them. The hardware can include channels, control units and devices (both ESCON and non-ESCON), ESCON Directors (they are not required), and hardware used for sysplex coordination such as coupling facilities and External Time Reference (ETR) devices.

Operator Terminals

SA OS/390 supports any display supported by Tivoli NetView for OS/390 Release 4. This is required for access to SA OS/390 system operations and processor operations functions through NetView.

SA OS/390 supports any display supported by ISPF 4.2 or higher. This is required for access to SA OS/390 I/O operations functions and the SA OS/390 customization dialogs.

Operating Systems Supported by Processor Operations

SA OS/390 processor operations monitors and controls target systems with the following operating systems:

- z/OS, OS/390, MVS/ESA, MVS/XA™ (MVS/SP 2.2 or higher)
- VM/SP 6.0, VM/XA 2.1, VM/ESA® 1.1.0
- VSE/SP 4.1, VSE/ESA™ 1.1.0 or higher
- LINUX of distributions providing Linux for zSeries and S/390 support

Note: The above products may no longer be serviced.

Supported Software

Integrated automation for the following products is supported:

Table 5. Supported Software

CICS TS Version 1.1 CICS TS Version 1.2 CICS TS Version 1.3 CICS TS Version 2.1	For integrated automation of CICS address spaces.
IMS Version 6 IMS Version 7 IMS Version 8	For integrated automation of IMS address spaces.
Tivoli OPC Version 2.3 Tivoli Workload Scheduler for z/OS Version 8	For integrated automation of OPC/TWS address spaces.
DB2 Version 6 DB2 Version 7	For integrated automation of DB2 address spaces.

Customization Dialog Considerations

The SA OS/390 customization dialogs do not provide National Language Support (NLS) in the ISPF environment. The SA OS/390 customization dialogs must be used with a terminal type of 3278. The terminal type can be set in the Terminal Characteristics portion of the ISPF settings panel.

Chapter 2. What is New in SA OS/390 2.2

This chapter contains an overview of the major changes to SA OS/390 for Version 2 Release 2. Use this information to check the impact on your user-written programming interfaces, such as automation procedures.

Changed Conceptual Product Behavior

This section introduces the most important changes in the behavior of the product in comparison to earlier releases.

OPC/TWS Changes

A new method of controlling System Automation from OPC/TWS has been introduced in this release. This new method will allow the OPC/TWS subsystem to issue any SA OS/390 or NetView command from a batch job. As a result of this new method of issuing SA OS/390 or NetView commands, the status of the resources affected by these commands will no longer be reflected in the status of the OPC/TWS operation that issued the commands. The OPC/TWS operation will be a normal batch job operation and will reflect the status of the batch job in the system. Return codes from the batch job can be used as in normal OPC/TWS operation definitions to determine the success or failure of the batch job.

Processor Operations Changes

The support for the following processors has been removed:

308X, 4381, 3090, 9021, 9121

The support for the operating system TPF has been removed.

The support for the SA OS/390 workstation and the processor operations PC has been removed. The following processor operations PC based commands are no longer valid:

```
EMULATE
END PASSTHRU
#DATA
#DISABLE
#ENABLE
#PORT START
#POWEROFF
#POWERON
#PURGE
#SCREEN
#SET
#SHARE
#SUPERRESET
#TIME
#QUERY
```

Message Automation Table Changes

SA OS/390 2.2 introduces a new Message Automation Table (MAT) structure. AOFMSG01 has been replaced by INGMSG01. For details about the new MAT structure, refer to *System Automation for OS/390 Customizing and Programming*.

Enhancements of Parallel Sysplex Operation

SA OS/390 2.2 provides functional enhancements that grant greater system reliability and availability in the sysplex. Sysplex management with SA OS/390 is enhanced as follows:

- A number of recovery actions can now be automated, for example:
  - Creating or re-creating missing alternate couple data sets (CDSs)
  - Expanding the system logger CDSs in case of a directory shortage
  - Resolving of WTO(R) buffer shortages
  - Avoiding sysplex outages
  - Resolving pending I/Os for systems being removed from the sysplex
  - Recovering auxiliary storage
  - Recovery of long-running ENQs

  Each automatic recovery action can be enabled or disabled separately. To do this, additional minor resources are introduced for the MVS Component policy object (MVC entry type). A number of these actions can be customized.
- The previously available sysplex-related commands have now been merged into two powerful commands, INGPLEX and INGCF. Furthermore, the following functions were introduced:
  - Making an alternate CDS the primary one
  - Defining a new alternate CDS
  - Switching the active policy
  - Integrating a coupling facility into the sysplex
  - Recording IPL information
  - Allowing for specifying dump options
  - Allowing for multisystem SVC dumps
  - Viewing, enabling disabling, and deleting SLIP traps defined in the sysplex.

For details about customizing and using these functional enhancements refer to *System Automation for OS/390 Defining Automation Policy*, *System Automation for OS/390 Operator's Commands*, and *System Automation for OS/390 User's Guide*.

This new edition incorporates APAR OA02321 that introduces the following changes and improvements:

- The IBM Health Checker for z/OS and Sysplex and its related command INGHC; for more information see *System Automation for OS/390 Operator's Commands*
- Improved resolution of pending I/Os for systems being removed from the sysplex
- Hardware validation

Changes with Commands

The following tables list the commands that are new, deleted, or changed in SA OS/390 2.2:

- Table 6 on page 11
- Table 7 on page 11
- Table 8 on page 12

Check your user-written CLISTSto find out whether these changes have an impact on them.

For a detailed and complete description of the SA OS/390 commands, refer to *System Automation for OS/390 Operator's Commands* and *System Automation for OS/390 Programmer's Reference*.

New Commands

Table 6. New Commands Shipped with SA OS/390 2.2

New Command	Short Description
INGHC	This command allows you to view the reports of the IBM Health Checker for z/OS and Sysplex (HealthChecker) and to request the HealthChecker to perform individual checks according to your filtering options.
INGRPT	This command displays statistical information about the automation agent and basic information about the automation manager.
INGTIMER	This command associates a NetView timer command with a subsystem.
INGLKUP	This command displays all active jobs and started tasks that are not controlled by SA OS/390.
INGCICS	This command allows the operator to interact with a CICS subsystem. It requires that CICS subsystems accept the MVS command MODIFY from EMCS consoles.
INGIMS	This command allows the operator to interact with a IMS subsystem. It requires that IMS is set up to accept commands via MVS MCS or EMCS consoles.
INGOPC/INGTWS	This command allows the operator to interact with the current plan of a Tivoli Workload Scheduler for z/OS controller. It requires that the TWS API is available to NetView.
INGVTAM	This command allows subsystems to register the use of VTAM APPLICATION IDs. SA OS/390 will then allow the user to specify recovery actions based on the status of VTAM or the registered nodes.

Deleted Commands

Table 7. Deleted Commands and their Replacements in SA OS/390 2.2

Deleted Command	Command with equivalent function (if applicable)
DISPCONS	INGPLEX CON
DISPPATH	INGCF PATH
DISPSTR	INGCF STR
DISPXSYS	INGPLEX SYS
DISPCDS	INGPLEX CDS
CFDRAIN	INGCF DRAIN
SETEVENT	INGEVENT
DISPATHR	INGTHRES
DISPTHRS	INGTHRES
@D	N/A
@DS	N/A
@V	N/A

Table 7. Deleted Commands and their Replacements in SA OS/390 2.2 (continued)

Deleted Command	Command with equivalent function (if applicable)
SHOWME	N/A
ISQTCC	N/A
ISQXACT	N/A
ISQXASN	N/A
ISQXCFC	N/A
ISQXDLC	N/A
ISQXDLP	N/A
ISQXFLS	N/A
ISQXFRC	N/A
ISQXFRM	N/A
ISQXGTC	N/A
ISQXIPL	N/A
ISQXLOD	N/A
ISQXPAR	N/A
ISQXPOR	N/A
ISQXSLP	N/A
ISQSWTCH	N/A

Enhanced Commands

Table 8. Enhanced Commands Shipped with SA OS/390 2.2

Command	Description
DISPFLGS	Invokes DISPSCHD and INGAUTO
IMSCMD	IMSCMD now uses the new INGIMS command to issue an IMS command against an IMS subsystem. As a consequence the results are no longer delivered using theEVI690I, EVI691I and EVI692I messages.However the IMS command prefix character (/) is now optional. Also the IMS subsystem can be a fully defined SA OS/390 APL resource.
INGPLEX	The BESTpractices parameter now allows you to view the currently active HealthChecker best practices.
INGTHRES	Combines the previous commands DISPATHR and DISPTHRS to display all defined thresholds. You can also set thresholds for a specific resource, or set the default for all resources.
ISQCCMD	New operations management commands and SNMP support

Changes with User Exits

User exits have been removed and new exits have been added to SA OS/390. The new user exits are documented in *System Automation for OS/390 Customizing and Programming*.

New User Exits:

- AOFEXC02

- AOCEXC03
- AOFEXC04
- AOFEXI03
- AOFEXI04

Chapter 3. Planning to Install SA OS/390 on Host Systems

Before you begin the post-SMP/E installation of SA OS/390, read *System Automation for OS/390 General Information*. That book briefly describes what SA OS/390 does and explains the concepts you need to know to operate the product. You also need to make sure that you or someone in your organization has done the SMP/E install.

Component Description

The SA OS/390 product consists of the following components:

- system operations, sometimes also abbreviated as *SysOps*
- processor operations, sometimes also abbreviated as *ProcOps*
- I/O operations, sometimes also abbreviated as *I/O Ops*

System Operations

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

Enterprise monitoring is used by SA OS/390 to update the NetView Management Console (NMC) resource status information which is stored in the Resource Object Data Manager (RODM).

Processor Operations

Processor operations monitors and controls processor hardware operations. It provides a connection from a focal point processor to a target processor. With

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

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

I/O Operations

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

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

SA OS/390 and Sysplex Hardware

When SA OS/390 is used in a Parallel Sysplex environment, the hardware setup can be similar to the one illustrated in Figure 1.

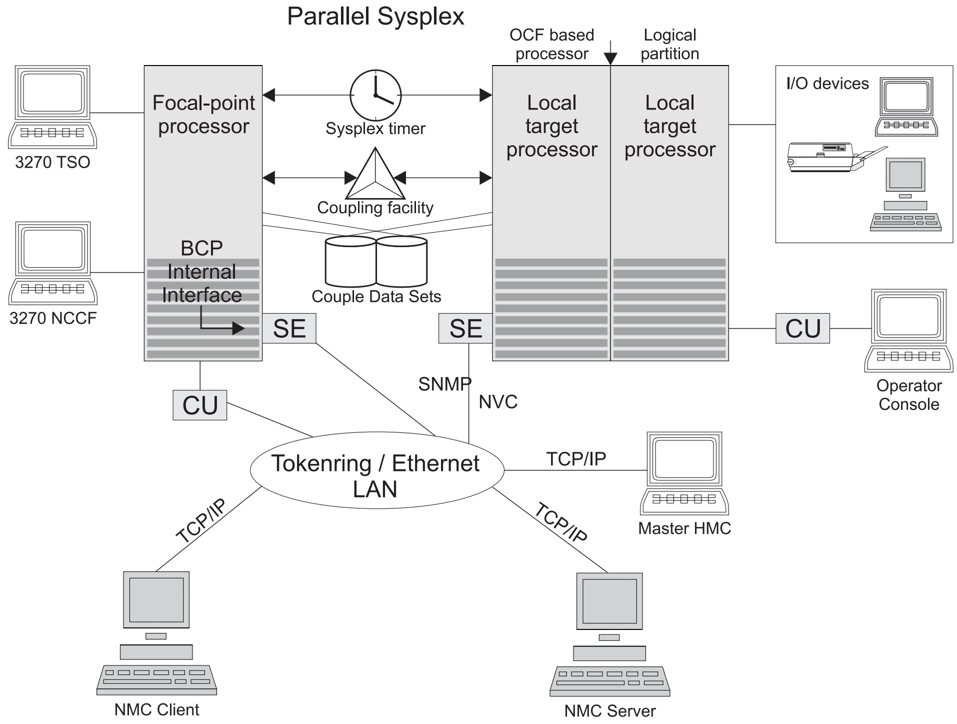


Figure 1. A Basic Hardware Configuration

It shows a two processor Parallel Sysplex configuration, with systems running on it. One is playing the role of a SA OS/390 focal point. For example, the role of the SA OS/390 NMC focal point with information about all the systems and applications in the sysplex, running under the control of SA OS/390.

Operators can use a workstation with the SA OS/390 NMC client code installed, to work with graphical views of the SA OS/390 controlled resources stored on the focal point. The NMC server component receives status changes from the NMC focal point and distributes them to the registered clients to update their dynamic resource views. Sysplex specific facilities, like the coupling facility hardware can be managed and controlled using the NMC's client graphical interface, as well as the 3270 NCCF based SA OS/390 operator interfaces.

With the same interfaces, processor operations, another SA OS/390 focal point function can be operated. With processor operations it is possible to manage and control the complete processor hardware in a sysplex. Operator tasks like re-IPLing a sysplex member, or activating a changed processor configuration can be accomplished. Processor operations uses the processor hardware infrastructure, consisting of the CPC Support Element (SE), or the Hardware Management Console (HMC) interconnected in a processor hardware LAN, to communicate with the own, other local, or remote located Support Elements of other CPCs. The Support Elements provide the Systems Management Interface OCF (Operations Command Facility) to perform hardware commands like LOAD or SYSTEM RESET to control the hardware and hardware images. SA OS/390 processor operations can be customized to use SNA based NetView connections (NVC), or IP based SNMP for communication. For Parallel Sysplex environments, SA OS/390 provides an additional processor hardware interface, the BCP (basic control program) internal interface. This interface is independent from processor operations. It allows processor hardware operation in a sysplex, without requiring external network CUs (control units). From a system in the sysplex, the SE of the own CPC as well as the SEs of the other processors in the sysplex can be accessed.

The following describes some relevant resources used by SA OS/390 and its components.

OCF-Based Processor

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

Parallel Sysplex

A set of OS/390 systems communicating and cooperating with each other through certain multi-system hardware components (coupling devices and sysplex timers) and software services (couple data sets). In a Parallel Sysplex, OS/390 provides the coupling services that handle the messages, data, and status for the parts of a multi-system application that has its workload spread across two or more of the connected processors. Sysplex timers, coupling facilities, and couple data sets containing policy and states for basic functions are all part of a Parallel Sysplex. You can control a Parallel Sysplex by NetView-based commands or through an NMC workstation.

Coupling Facility

A hardware storage element with a high-speed cache, list processor, and locking functions that provides high performance random access to data for one system image or data that is shared among system images in a sysplex. With I/O operations you can see stand-alone coupling facilities. It handles them as control units with up to eight devices, all defined by the user. With SA OS/390 system operations, you can display the status of coupling facilities from a single system's point of view or you can display sysplex-wide status.

Sysplex Timer

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

Logically Partitioned (LPAR) mode

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

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

Communications Links

Links that connect the focal point processor to target processors so that commands, messages, and alerts can flow. For more information refer to “Defining System Operations Connectivity” on page 45.

NetView Connection (NVC)

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

SNMP

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

BCP Internal Interface

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

NetView to NetView

A connection through which status changes are routed from the system operations target system to the system operations focal point.

Control Units (CU)

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

I/O Devices

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

NetView Management Console (NMC)

A NetView function that consists of a graphic series of windows controlled by the NetView program and that allows you to monitor the SA OS/390 enterprise interactively. The NetView Management Console consists of an NMC server and an NMC client.

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

Planning the Hardware Interfaces

This section provides additional information about the processor hardware interfaces supported by SA OS/390.

Understanding the BCP Internal Interface

In order to allow the sysplex-wide activation or deactivation of the coupling facilities and to control sysplex members leaving the sysplex, SA OS/390 uses the BCP (Basic Control Program) internal interface. The BCP Internal Interface of the following processor hardware families is supported:

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

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

Furthermore, with the enhanced sysplex functions of SA OS/390, sysplex members running on other CPCs than their own image can be controlled through the BCP Internal Interface. This is possible by defining all CPCs of your sysplex on the master HMC of your processor hardware LAN.

The following processor hardware can be controlled as a target with the BCP Internal Interface of the above listed processors, but cannot use the SA OS/390 BCP Internal Interface to control itself or other processors:

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

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

Note that the MVS/HCD function uses the BCP Internal Interface to update IOCDS and IPL information in the Support Elements of addressed CPCs. You cannot use SA OS/390 to perform these tasks, nor can HCD be used to perform the hardware operations functions of SA OS/390.

Currently, the BCP Internal Interface cannot be used by the processor operations focal point application. The interface can be configured and used for Parallel Sysplex automation purposes only.

Understanding the Processor Operations SNMP Interface

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

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

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

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

The Support Elements of the CPCs you want to control must be configured for SNMP. Alternatively, you can configure a single HMC instead of multiple Support Elements in your processor LAN environment for SNMP. On this HMC the CPCs you want to control must be defined. Multiple HMCs, SEs, or both can be defined in your SA OS/390 configuration.

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

Understanding the NetView Connection (NVC) of Processor Operations

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

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

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

Deciding which Hardware Interface to Use

If you want to use the Parallel Sysplex enhancements of SA OS/390 and you have configured your customization to use the XCF automation, the BCP Internal Interface is required.

Note, that this interface can coexist with the supported SNMP and NVC interfaces on a processor operations focal point system. Because the IXC102A automation, which is part of the Parallel Sysplex XCF automation, can also be performed in SA OS/390 using proxy resources together with processor operations, a decision must be made, which automation to use. It is recommended to use the XCF automation based on the BCP Internal Interface and to disable the IXC102A proxy resource automation based on processor operations.

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

- Processor hardware LAN
- Processor hardware type

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

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

Using SA OS/390 Partitioned Data Sets

You must be aware of the restrictions involved in the use of partitioned data sets by SA OS/390 and NetView. This section provides information on:

- Allocating SA OS/390 partitioned data sets
- Using LNKLSTxx (Link Library List)
- Sharing data sets

Allocating SA OS/390 Partitioned Data Sets

After you have completed the SMP/E installation of SA OS/390, you need to allocate data sets to hold your locally customized members. These data sets are concatenated in various data definition (DD) names in the NetView startup procedure. You may need data sets for the DSIPARM, DSIMSG, DSICLD, CNMPNL1, STEPLIB, and DSIPRF concatenations.

There are two types of customization that you may perform:

- Enterprise-specific
- Domain-specific

Domain-specific Customization

SA OS/390 DSIPARM

This data set holds SA OS/390 NetView definitions that are specific to this NetView. You may need it only for the DSIPARM concatenation. It should contain the CNMSTYLE specifications that define this domain and a suitably customized AOFMSGSY member. For your gateway definitions and your alert forwarding related to NMC, the updated DSIDMNK member should be included. It is recommended to use NV-UNIQ forwarding. According to your NMC setup, it should contain your NMC specifications like the INGTOPOF file and the corresponding BLDVIEWS members. For information about this customization, see *System Automation for OS/390 Customizing and Programming*.

Enterprise-specific Customization

SA OS/390 ACF

This data set holds the various automation control file fragments and control file fragments for the automation manager needed by the SA OS/390 instance on this NetView. It is not recommended that you build your SA OS/390 policy databases directly into this data set. Changing your active SA OS/390 policy is a process that should be under change control.

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

Notes:

1. If the automation agent and the automation manager use different data sets they must have the same name.
2. INGAMS refreshes GDG data sets by resolving these to absolute data set names during dynamic allocation.
3. It is highly recommended that the ACF data set is not placed into the DSIPARM concatenation. The reasons for this are as follows:
   a. The same data set must be specified to the automation manager.
   b. The DSIPARM data set cannot be changed without stopping and starting all agents within the sysplex.
   c. If you use INGAMS to refresh your configuration and you specify another data set, dynamic allocation will take place, the data set allocated to DSIPARM is ignored. The control file you see with NetView browse will not match the control file used by the automation manager.
   d. GDG versions are resolved at runtime by the automation manager, while the automation agent receives the complete data set name at NetView startup. If a different GDG version is allocated to DSIPARM, dynamic allocation takes place. Again, the control file you see with NetView browse will not match the control file used by the automation manager.
4. If you use MEMSTORE to load the NetView PDS members in storage (this is the default in NetView 1.4) and you do not reload the members after an ACF build, you will get message `AOF618I ... ACF Token mismatch ...` due to an INGAMS command to refresh the configuration.

SA OS/390 DSIPARM

This contains SA OS/390 NetView definitions that you have modified and that are applicable to all your systems. This is required on all NetView instances running SA OS/390. This data set contains SDF definitions, and possibly modified NetView message automation table fragments.

SA OS/390 CREXX

Compilation of SA OS/390 REXX code is optional and supported. If you choose to compile the CLISTs in the SA OS/390 SMP/E target data set, this is where the compiled versions should be placed. It is needed only in the DSICLD concatenation for system operations and processor operations REXX CLISTs. If you have compiled I/O operations execs, then the data set with these compiled execs must be in the SYSEXEC concatenation.

Note: If you use this data set, you can omit the SA OS/390 SMP/E target data set in your DSICLD concatenation, because all the code is in it as well.

SA OS/390 Libraries
This contains data set members shipped with SA OS/390. It can either be the real SMP/E data set or a locally made copy. If you must change the contents of any SA OS/390 part, copy the member from the SMP/E target data set to an SA OS/390 enterprise-specific or domain-specific data set, and edit it there. By keeping the contents of this data set as shipped, you make it much safer to apply SMP/E maintenance to its members, because SA OS/390 fixes will not overwrite your enterprise-specific modifications.

Using LNKLSTxx (Link Library List)

If you have put the NetView and SA OS/390 data sets into your LNKLST concatenation, rather than into STEPLIB, they must be in the following sequence: the SA OS/390 ones before the NetView ones.

Sharing Data Sets

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

One solution is to have a set of standby procedures for SA OS/390. These are copies of your normal SA OS/390 procedures that point to a copy of your data set on another DASD volume and preferably on a different string of DASD volumes. Although an instance of SA OS/390 started from these procedures would not share status information with the SA OS/390 from your primary procedures, the standby procedures let you maintain operability of your systems in the event that your primary procedures are unavailable.

SA OS/390 Password Protection Feature

The SA OS/390 password protection is an optional feature that allows SA OS/390 to use an SAF program, such as RACF® or a functionally equivalent product, to manage gateway operator IDs and passwords. This feature is intended for systems that establish gateway sessions with two or more other systems.

The SA OS/390 password protection feature involves the SA OS/390 password data set and the SAF program. The SA OS/390 password data set contains only gateway operator ID and password entries. The SAF program manages all user IDs and their passwords including gateway operator IDs and their passwords.

Consider the following if you plan to install the SA OS/390 password protection feature:

- For an automation network to use the SA OS/390 password protection feature, all systems in the network must install the feature.

- An SAF program must be installed on all systems that are to use the SA OS/390 password protection feature.
- The number of gateway sessions each system in your automation network establishes with other systems.
- You need to decide whether each system in your automation network is to have its own SAF product data set or whether several systems will share one password data set. See *System Automation for OS/390 Defining Automation Policy* for more information about shared SAF data sets.

To install the SA OS/390 password protection feature, install SA OS/390 as described in “Step 20: Define Security” on page 110.

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

REXX Considerations

Allocation Requirements for REXX Environments

Before running SA OS/390 you may need to change the maximum number of REXX environments allowable.

The recommended starting point is 400 concurrent REXX environments for the address space of your SA OS/390. Different SA OS/390 configurations may require more REXX environments.

The number of REXX environments allowable is defined in the REXX environment table. See *OS/390 TSO/E Customization* for more information. TSO/E provides a SYS1.SAMPLIB member called IRXTSMPE, which is an SMP/E user modification to change the maximum number of language processor environments in an address space. Define the number of allowable REXX environments on the IRXANCHR macro invocation:

```
IRXANCHR ENTRYNUM=401
```

Install the user modification by following the instructions in *OS/390 TSO/E Customization*.

Changing NetView REXX Environment Usage Characteristics

It is recommended that you change some of the NetView REXX environmental usage defaults. This can be done using the NetView DEFAULTS or OVERRIDE commands with the REXXENV and REXXSMLT parameters.

REXXENV

This is the number of REXX environments that each task “hangs onto” when it has finished with them. The default is 10, but 3 usually gives quite satisfactory performance.

REXXSMLT

This is the limit on the total amount of storage that NetView is allowed to allocate for REXX environments. The default is unlimited, but 500K usually gives satisfactory performance. If you use many locally written REXX, you may wish to use a larger value.

See *Tivoli NetView for OS/390 Automated Operations Network User’s Guide* for details of the DEFAULTS and OVERRIDE commands.

OS/390 Considerations

SYS1.PARMLIB Member Suffix

The xx suffix on each SYS1.PARMLIB data set member can be any two characters chosen to match your IEASYS naming scheme. Before starting installation allocate a suffix for SA OS/390, checking that this suffix is not already in use. See *z/OS MVS Initialization and Tuning Reference* for information about IEASYS.

Defining the XCF Group

In order to be able to communicate in certain situations, the automation manager instances and the automation agents belonging to one sysplex must be members of one and the same XCF group. The name of this group consists of a fixed main part and a variable suffix; the format is INGXSG*xx*. The suffix must be specified separately for the manager and the agents. For the automation manager, it is specified in the HSAPRM*xx* member of SYS1.PARMLIB (see “Step 9A: Customizing HSAPRM*xx*” on page 89); for the automation agents, it is defined in the INGXINIT member of DSIPARM (see “INGXINIT” on page 80).

Systems with SA OS/390 NetView instances belonging to the same XCF group must be defined in the Customization Dialogs in the same Group Policy Object of type sysplex. For details refer to the chapter ″Group Policy Object″ in *System Automation for OS/390 Defining Automation Policy*.

Note that SA OS/390 NetView instances belonging to the same XCF group must reside on different systems. Thus, when you run an SA OS/390 V2R1, or SA OS/390 V2R2 agent and an instance of msys for Operations on the same system, they must not belong to the same XCF group.

Using SA OS/390 Subplexes

You can divide your real sysplexes into several logical SA OS/390 subplexes. To do this you must define a specific XCF group suffix and a specific group policy object for each subplex. Each SA OS/390 subplex must have an own automation manager running. Using SA OS/390 subplexes you can run the automation on systems of sysplexes like on single systems. This is required if you do not have shared DASDs for all your systems in the sysplex.

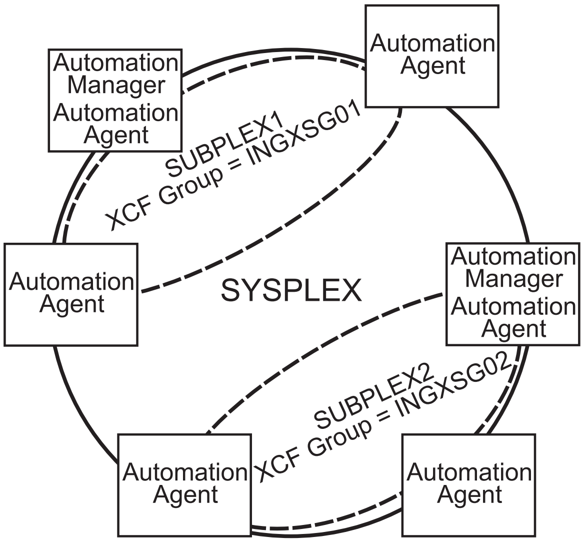


Figure 2. Using SA OS/390 Subplexes

Planning XCF Transport Classes

By explicitly assigning the XCF group `INGXSGxy` to a transport class, you give the group priority access to the signalling resources (signalling paths and messages buffer space) of the transport class.

The person responsible for the sysplex setup needs to know that SA OS/390 has a maximum XCF message length of 3500 bytes. Either an existing transport class with the appropriate class length may be used, or a new transport class can be defined. In either case the group name `INGXSGxy` needs to be associated with this transport class (see *z/OS MVS Setting Up a Sysplex*).

NetView Considerations

NetView ships two sample automation operators, AUTO1 and AUTO2. SA OS/390 assumes that these tasks are available and have not been renamed. If they have been renamed, you must change the names in AOFMSGSY and CNMSTYLE residing in the DSIPARM data set.

In CNMSTYLE, change every occurrence of AUTO1 and AUTO2 to the autotask names you are using.

```
AUTOTASK.AUTO1.Console=*NONE*
AUTOTASK.AUTO1.InitCmd=AOFRANTL AUTO1,AUTO2
function.autotask.primary = AUTO1

AUTOTASK.AUTO2.Console=*NONE*
AUTOTASK.AUTO2.InitCmd=AOFRAAIC
```

It is recommended that you have system operations and processor operations installed together on the status focal point system.

Automation Manager Considerations

This section presents automation manager considerations relevant to the installation process. For automation manager concepts that are interesting from an operator's point of view, refer to *System Automation for OS/390 User's Guide*.

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

A series of substeps is required to get the automation manager up and running for your SA OS/390 installation. These installation steps are described in this documentation, but are not identified as being specific automation manager installation steps.

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

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

Note: The system on which the automation manager should be started must be defined as policy object *System* in the policy database which will be used to create the automation manager configuration file that this automation manager uses (see also "Step 16A: Build the Control Files" on page 106.

Considerations when using MIM instead of GRS:

If you are using MIM instead of GRS, be careful with mimplexes that extend beyond the sysplex. Mimplexes that extend beyond the sysplex can cause automation manager initialization to fail due to ENQ contention. Using different GRPIDs for each sysplex within the mimplex can resolve this problem.

Storage Requirements

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

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

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

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

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

OMVS Setup

Because the Automation Manager requires OMVS, OMVS must be customized to run without JES2. Therefore the definitions in the BPXPRM*xx* member must match *one* of the following:

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

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

Recovery Concept for the Automation Manager

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

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

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

To enable software or hardware maintenance in the sysplex, SA OS/390 supports a command to force a takeover of the primary automation manager.

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

- All the automation manager instances must have access to a shared external medium (DASD) where the following is stored:
  - the configuration data (result of the ACF and AMC build process)
  - the schedule overrides VSAM file
  - the configuration info data set which is a mini file in which the automation manager stores the parameters with which to initialize the next time when it is started WARM or HOT.
  - the takeover file
- If MQSeries is used for communication between the automation manager and the automation agents (see "Manager-Agent Communication and Status Backup" on page 29), all the automation manager instances must have access to the coupling facility that contains the status backup queue.
- If MQSeries is used for communication between the automation manager and the automation agents, the automation agents must have access to the coupling facility that contains the agent and the workitem queues

SA OS/390 follows the concept of a floating backup, because:

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

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

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

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

Figure 3 shows the configuration of automation manager recovery. The data required for a restart are externalized on a shared I/O device.

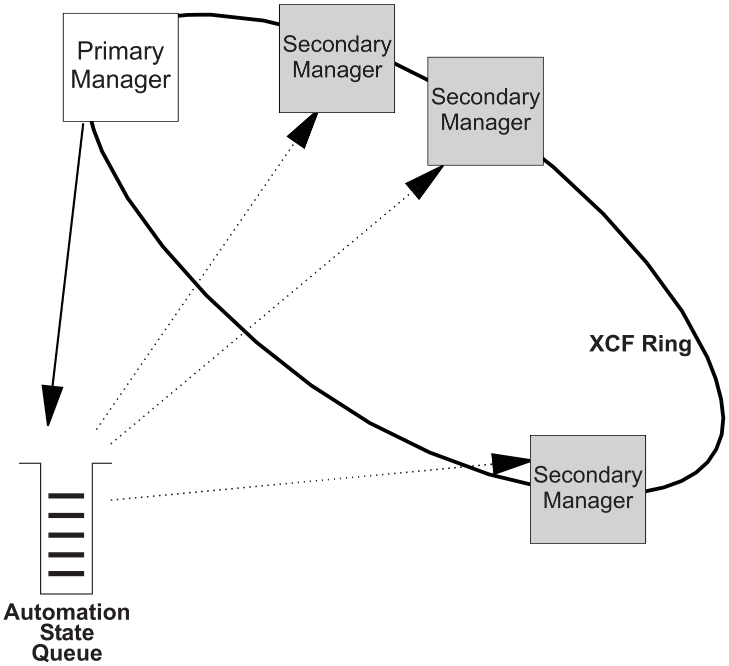


Figure 3. Recovery Concept of the Automation Manager when using MQSeries

Manager-Agent Communication and Status Backup

SA OS/390 provides two options for establishing communication between the automation manager and the automation agents and keeping a backup copy of the status of the automated resources:

- Using MQSeries V5R2 queues for both communication and backup
- Using XCF for communication and a VSAM data set for backup

For the MQSeries solution, you need MQSeries V5R2 or higher, and also DB2 in data sharing mode. V5R2 is required because the PAM and the SAMs must share the queues; this data sharing capability, which is implemented by means of coupling facility list structures, is shipped with MQSeries V5R2. DB2 is required because it serves as the repository for the definitions of the shared queues.

Note: SA OS/390 also supports a local system environment with a MQSeries V2R1. In that case, DB2 is not required, but the scope of automation is limited to a single system.

Exploiting MQSeries V5R2

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

- **Workitem Queue:** This queue is the inbound queue for the automation manager. The automation agents put their requests/queries in form of a workitem into this queue. Its name is WORKITEM.QUEUE.
- **Agent Queue:** This is the outbound queue of the automation manager. All orders for the automation agents that result from a request (a workitem) sent to the automation manager are placed in this queue by the automation manager. The automation agents are then picking up the orders from this queue for execution. Its name is AGENT.QUEUE.
- **Automation State Queue:** This queue is only used by the automation manager. It is used to save the current state as well as other information about the resources managed by the automation manager. It is the **automation state queue** that allows SA OS/390 to perform a hot takeover, since this queue contains at any point in time a consistent image of the resource data that the PAM maintains in storage. Any updates made to the resources are also reflected in the **automation state queue**. Its name is STATE.QUEUE.

The transactional behavior of MQSeries ensures that these three queues are always consistent. A change in any queue is only committed after the corresponding changes have also been made in the other two queues. Thus, for example, the deletion of a workitem from the **Workitem Queue** is only committed when the resulting orders to the agents have been written to the **Agent Queue** and the resulting state changes of the affected resources have been written to the **Automation State Queue**.

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

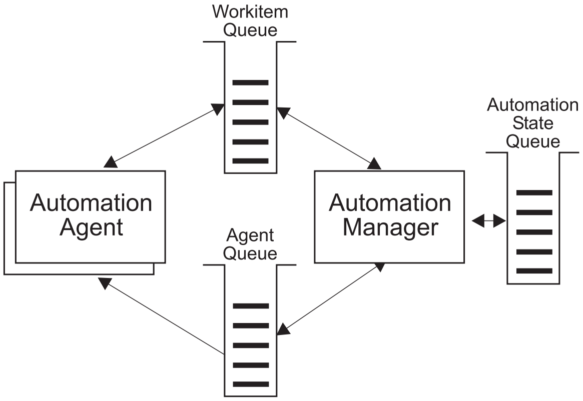


Figure 4. MQSeries Queues

The three queues are shared between the PAM and the secondary automation managers (SAMs). When the PAM fails, an SAM becomes the new PAM and takes over the shared queues in a consistent state. Since not only the actual state of the automated resources, but also the unprocessed requests (workitems) – even those that were made during the takeover phase – and all unprocessed orders for the automation agents are known, no requests are lost, only the execution time can be delayed.

When you use MQSeries for manager-agent communication and status backup, it is recommended that you automate MQSeries and let it be started and stopped by SA OS/390. Collecting all the MQSeries and DB2 instances in a basic group allows you to monitor these prerequisites of SA OS/390 as you would do with normal application automation.

Automation of MQSeries by SA OS/390 implies that MQSeries is not yet available when the automation manager is started. In this situation, the automation manager will communicate with the automation agents through XCF services until it has started its local MQSeries. This phase is called the *startup phase*. As soon as MQSeries is up, the automation manager switches over to MQSeries for communication with the automation agents and for status backup. This means that both the SA OS/390 automation manager and automation agents are now MQSeries applications.

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

During these ″MQSeries-less″ phases, resources can change their status, and the information about these changes should be preserved for an eventual successor of the actual PAM. This applies not only to a shutdown and subsequent restart of SA OS/390, but also to a failure of the PAM and a subsequent takeover by a SAM during the startup phase. To this end, SA OS/390 maintains a *takeover file*.

Every status change during the startup and shutdown phase of SA OS/390 is recorded in the takeover file. The information in this file is kept consistent by maintaining two compartments for each resource record. These compartments are used alternately to store the changes. This ensures that a consistent and fairly actual version of the resource information exists even when the PAM fails in the middle of an update of the takeover file.

When the PAM fails during startup, a SAM becomes the PAM as before. But now, the new PAM reads the takeover file and starts with the information contained therein. When the shutdown phase is terminated normally or abnormally, the takeover file will be used for a restart of SA OS/390. In this way, a hot takeover or restart is possible even when MQSeries is not available. Of course, this requires that the takeover file be shared between the PAM and the SAMs.

The following figure shows the time line for the MQSeries solution when MQSeries is automated by SA OS/390:

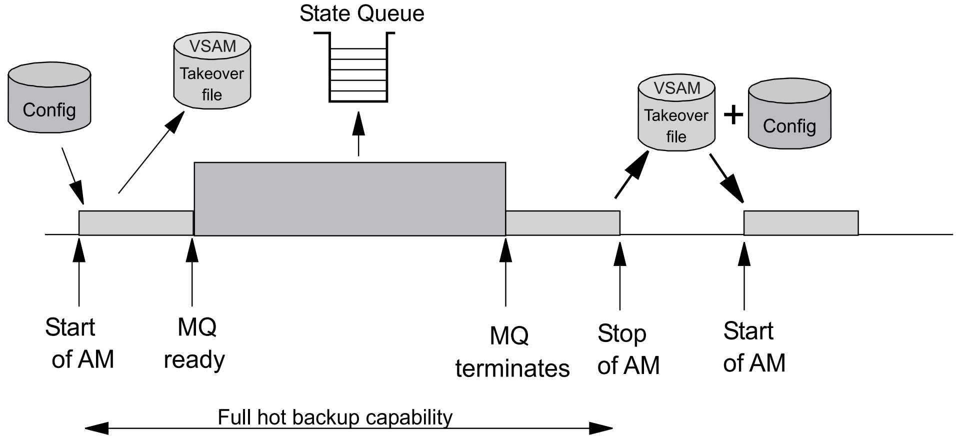


Figure 5. Automating MQSeries with SA OS/390

The sequence of events is as follows:

1. During the startup phase, when MQSeries is not yet running, the PAM uses XCF for communication and stores every status change in the takeover file.
2. (Not represented in Figure 5) If the PAM or its system fail during the startup phase, a SAM becomes the new PAM. The new PAM reads the actual state of the automatable resources from the takeover file.
3. As soon as MQSeries is up, the PAM switches to MQSeries and writes the status updates into the **Automation State Queue**.
4. (Not represented in Figure 5) If the PAM or its system fail while MQSeries is running, a SAM becomes the new PAM. The new PAM uses the information of the shared MQSeries queues.
5. When the PAM has shut down its own local MQSeries, the reaction of SA OS/390 depends on whether an SAM is available:
   - (Not represented in Figure 5) When SAMs exist one of these becomes the new PAM.
   - When no SAM exists, the current PAM enters the shutdown phase. It switches back to XCF and the takeover file.
6. During the shutdown phase, the PAM stores every status change in the takeover file. After the PAM has terminated normally or abnormally, the information of the takeover file will be used for a restart of SA OS/390.

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

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

The following subsection contains an overview of various recovery scenarios.

Some Problem Scenarios and how SA OS/390 Reacts: Some examples of how SA OS/390 reacts when there is a:

system breakage with running automation manager and automation agent
A waiting secondary automation manager will automatically take over the responsibility of the failed primary automation manager. Of course the broken automation agent is not moved, because all the broken resources are gone anyway. However the new automation manager will detect the system collapse and react accordingly.

an automation manager breakage
A waiting secondary automation manager or the ARM restarted primary automation manager will automatically take over the responsibility.

an MQSeries manager breakage
A connected automation agent will wait for the ARM initiated MQSeries manager restart. A connected automation manager will automatically trigger a takeover in case of active processes, otherwise this automation manager would also wait. Please note that even in the case of an MQSeries manager abend or problem, the automation agent is still able to do message automation.

a DB2 problem or breakage
After DB2 first came up and SA OS/390 was able to access the MQSeries queues the first time, DB2 is actually not needed any longer. Therefore any of these cases can be completely automated even in the full MQSeries supported fashion.

a CF outage
At this point in time, SA OS/390 will automatically restart its automation processing from the existing static configuration (WARM start).

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

MQSeries Queue Problems
Please see chapter “MQSeries Exception Processing” on page 34.

MQSeries Considerations

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

Peer Recovery Considerations: Please refer to the MQSeries V5.2 documentation for all aspects of MQSeries sysplex-wide peer recovery. Because SA OS/390 exploits this technology, you also gain this functionality. The following are important things that you should consider when planning for SA OS/390 to be an MQSeries shared queues exploiter.

The basic setup consideration is whether you choose to have a dedicated MQSeries® QSG (Queue Sharing Group) just for the purpose of SA OS/390.

Peer recovery requires that a failed MQSeries instance should be restarted in place either using OS/390 Automatic Restart Manager or SA OS/390 itself.

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

SA OS/390 will ensure that all the pending work has been rolled back before the new primary automation manager starts accessing the queues.

There are no special considerations for DB2 in case of a takeover. DB2 is not involved in the MQSeries peer recovery functions.

Now since SA OS/390 provides the capability for automating its prerequisites, SA OS/390 could be used to initially start MQSeries and DB2. OS/390 Automatic Restart Manager or SA OS/390 could be used to restart the MQSeries instances and SA OS/390 can be used to observe the status of its prerequisites as well as to finally stop it.

Collecting all the MQSeries and DB2 instances in a basic group allows you to monitor all SA OS/390 prerequisites as you would do with normal application automation.

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

recoverable errors
can be recovered by running the failed MQSeries service again after a certain time.

unrecoverable errors
cannot be recovered automatically.

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

The automation manager will trigger a takeover in case there is a local MQSeries manager problem with active transactions.

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

- re-define the Queue on a different CF
- re-define the Queue on a different CF Structure
- increase the number of maximum possible messages.

The current automation manager will continuously monitor whether there is a new Automation State Queue with both GET and PUT enabled again. If this automation manager finds a queue with that attribute it will try to re-install the queue. Having done this, SA OS/390 is fully recoverable again.

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

Recovery from a queue full condition will be automatically done by the current active SA OS/390 automation manager. It can be considered as a recoverable exception as described above.

Situations where a queue full condition can occur are:

1. For the Workitem Queue
   - an automation manager is not available to pick up the workitem requests (for example, just stopped or restarting)
   - an automation agent based automation CLIST repetitively sends requests to the automation manager. Since these automation manager requests can be generated by an automation program, this program may loop.
2. For the Agent Queue
   - automation agent(s) are not available or able to process orders or responses in time.
   - Many concurrent big response blocks
3. Automation State Queue
   - Dynamic Configuration Reloads increase the amount of Automation State information drastically

Taking the SA OS/390 delivered samples together with a healthy system should not result in such a situation. It is more an indication that something is wrong but SA OS/390 tries its best to survive.

The recovery action for an Automation State Queue full condition is described in "MQSeries Exception Processing" on page 34.

For the Automation Workitem Queue as well as the Automation Agent Queue, SA OS/390 introduces three additional message counters to act as thresholds:

Low Threshold
: if the numbers of messages are below that line, operations on that queue are fine, no recovery actions are taken.

High Threshold
: if the number of current active messages reaches that count, recovery actions are taken. To stop this recovery mode again the number of messages must fall below the low threshold counter.

Max_Queue_Depth
: at this time, further MQPUTs are rejected, however SA OS/390 would retry.

Use the INGAMS command described in *System Automation for OS/390 Operator's Commands* to see these thresholds.

Using XCF only

XCF communication and the takeover file are primarily intended to substitute for the MQSeries queues in certain situations. However, you can also use them permanently and thus dispense with MQSeries altogether. If you wish to do so, you must set the COMM parameter in the HSAPRM*xx* member accordingly; see Appendix F, "Syntax for HSAPRM00", on page 189.

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

The following figure illustrates the time line from the start of the automation manager (AM) until its termination for both cases: a planned stop/start of the automation manager and an unexpected failure.

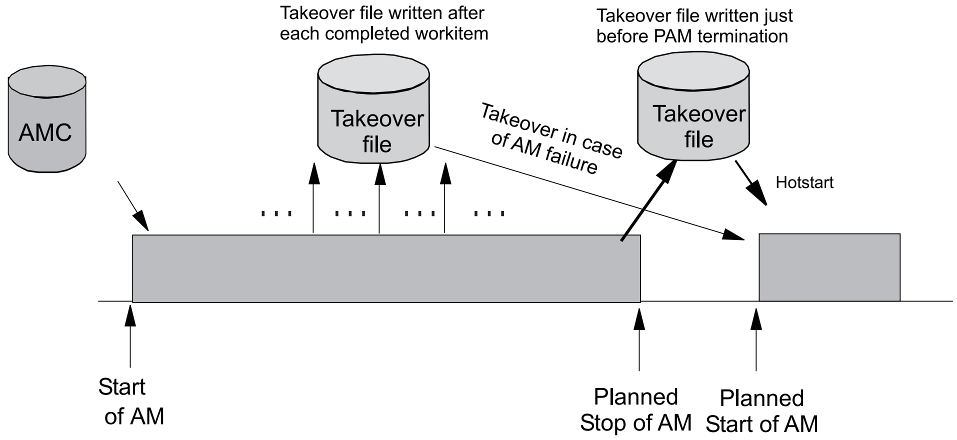

Takeover file written after each completed workitem
Takeover file written just before PAM termination
AMC
Takeover file
Takeover in case of AM failure
Takeover file
Hotstart
. . .
. . .
. . .
. . .
Start of AM
Planned Stop of AM
Planned Start of AM

Figure 6. Using only the Takeover File for Status Backup

The following table outlines the various recovery scenarios:

Table 9. Recovery Scenarios

Event	SA OS/390 Recovery Action	Comments
PAM fails	SAM runs a takeover	The takeover file contains the state with the last successfully processed workitem
PAM detects a severe error condition	PAM terminates and SAM runs a takeover	The takeover file is used to rebuild the resource object structures in case of a takeover or next hotstart.
System with the PAM fails	SAM runs a takeover	The takeover file is used to rebuild the resource object structures in case of a takeover or next hotstart.

Chapter 4. Planning to Install TEC Notification by SA OS/390

This section contains information required for the installation of TEC Notification by SA OS/390.

Introduction of TEC Notification by SA OS/390

The SA OS/390 notification is extended to notify Tivoli Enterprise Console™ (TEC) about an automation problem on OS/390 by sending an event to the TEC event server.

For this purpose, on OS/390 systems, messages or alerts are transformed into Tivoli events and sent to the ***TEC event server*** which is running on a Tivoli managed node in your network.

These events in turn may cause a notification of a Tivoli administrator on the TEC, may be correlated with other events on the TEC event server, or may result in opening a trouble ticket for example, dependent on what you programmed at the TEC event server.

SA OS/390 introduces the TEC event server as a new notification target for SA OS/390. Note that not all messages, but only messages indicating critical situations and alerts are forwarded as TEC events to the TEC event server using the respective Tivoli GEM Event/Automation Service Adapter.

A Tivoli administrator who wants to deal with a problem indicated by an event forwarded to the TEC event server by SA OS/390 needs access to the affected OS/390 system. For this purpose, you may use the Tivoli NetView 3270 Management Console. With TEC Notification by SA OS/390, the TEC administrator may log on to the NetView operator console by starting the NetView 3270 Management Console from the TEC console by executing a task. See *System Automation for OS/390 User's Guide* for a description of the graphical interface on how to achieve this.

Note: Forwarding of SA OS/390 messages to TEC will not start until SA OS/390 and the GEM Event/Automation Service are up and running. SA OS/390 messages issued during SA OS/390 startup will not be forwarded to TEC.

Environment Configurations

Several products are involved in TEC Notification by SA OS/390:

- ***Tivoli NetView for OS/390***
- ***Tivoli Global Enterprise Manager*** (GEM)
- ***Tivoli Enterprise Console*** (TEC)

You can run TEC Notification by SA OS/390 in two configurations:

- ***local configuration***: The message adapter or alert adapter is running on the same OS/390 system on which SA OS/390 is also running. The adapters are local to the SA OS/390 which is issuing and forwarding messages and alerts to the Tivoli Enterprise Console. Such a configuration for message forwarding is illustrated in Figure 7 on page 38.

- *distributed configuration*: The message adapter or alert adapter is running on an OS/390 system different from the one on which SA OS/390 is running and issuing messages and alerts. In this scenario, the OS/390 system running the adapters must be the SA OS/390 automation focal point system. Such a configuration for message forwarding is illustrated in Figure 8 on page 39.

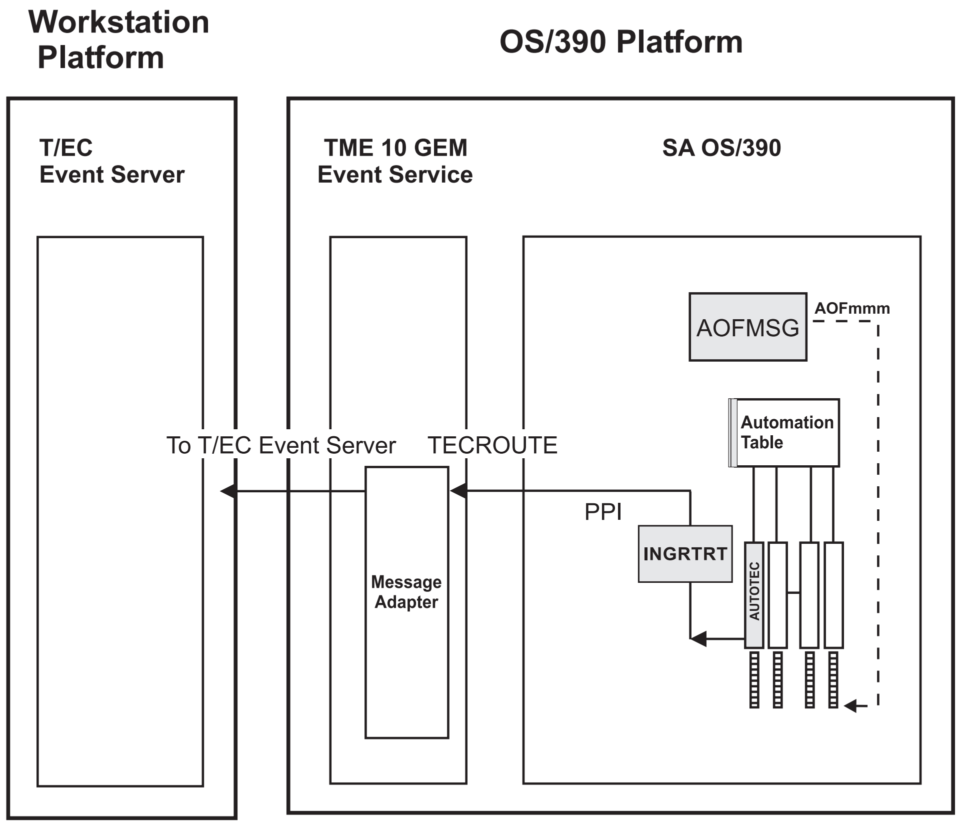


Figure 7. Local Configuration: Tivoli GEM Event/Automation Service local to the SA OS/390 Source of Messages

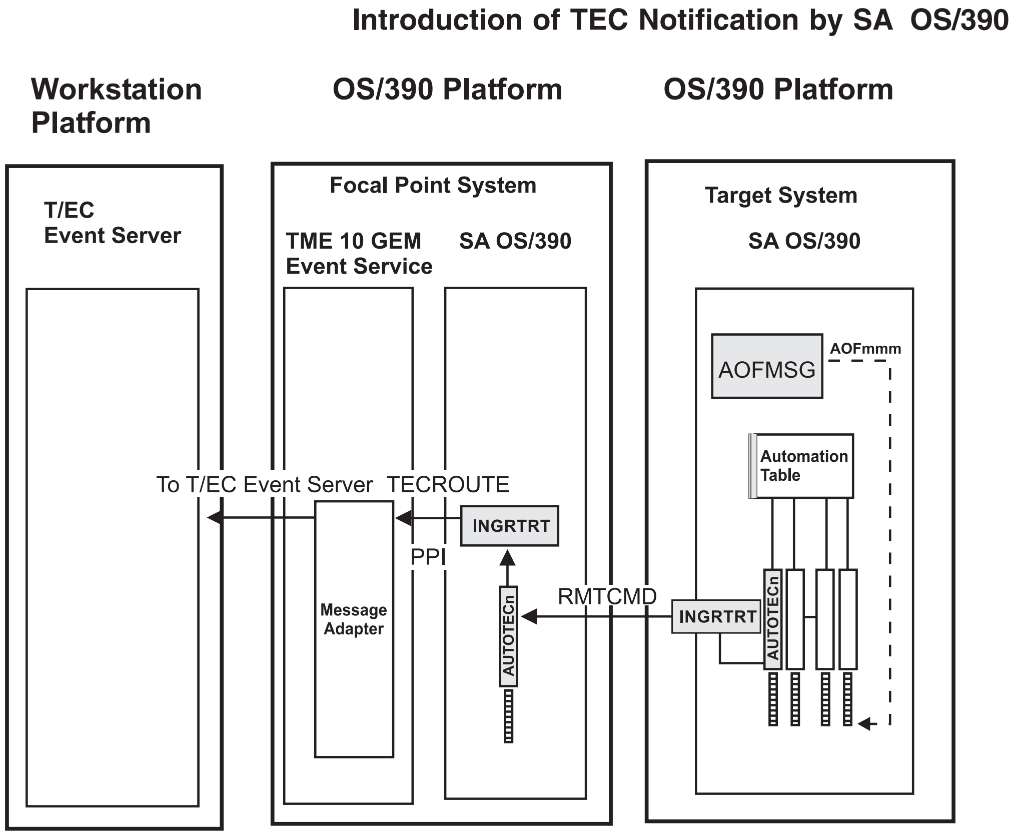


Figure 8. Distributed Configuration: Tivoli GEM Event/Automation Service remote to the SA OS/390 Source of Messages

Section “Environment Configurations” on page 37 describes the two configurations in which you can run TEC Notification by SA OS/390. How to install this feature is described in the following sections:
- “Installing and Customizing the TEC Event Server Workstation” on page 135
- “Activating the Installed Files” on page 136

The customization part comprises the following steps:
- “Step 14: Customization of NetView for TEC Notification by SA OS/390” on page 102 describes how to customize your SA OS/390 and TEC Notification by SA OS/390 installations on the OS/390 system for both the local and distributed configuration as described in “Environment Configurations” on page 37.
- “Customization of the Tivoli Enterprise Console” on page 137 describes how to customize and define event sources, event groups and how to assign event groups to a Tivoli administrator’s event console.

You can find more conceptual information about TEC Notification by SA OS/390 and information on how to use it in *System Automation for OS/390 User’s Guide*.

Chapter 5. Planning for the NMC Environment

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

NMC Exploitation Topology

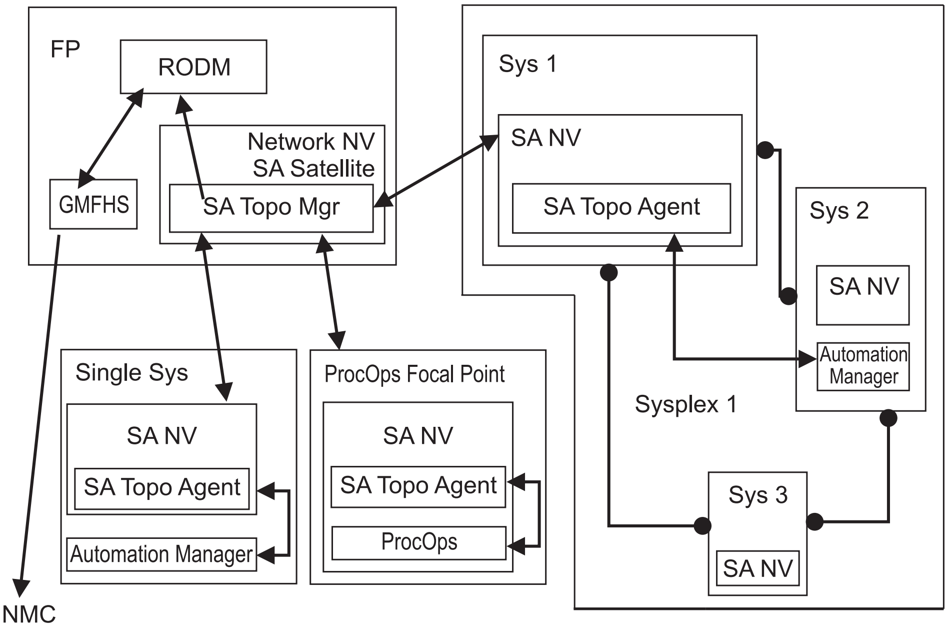


Figure 9. The SA OS/390 Environment for NMC Support

Figure 9 shows how in a SA OS/390 configuration the involved components communicate to produce graphical output information:

1. At initialization time, the SA OS/390 topology manager knows the target systems for automation.
2. The SA OS/390 topology manager contacts the SA OS/390 topology agents on all sysplexes or stand-alone systems or, for processor operations, it contacts the processor operations focal point to obtain the required information.
3. The SA OS/390 topology agents contact the related automation managers or the processor operations component respectively to find out the status from the systems and resources.
4. Then the SA OS/390 topology agents report this information to the SA OS/390 topology manager on the focal point.
5. The SA OS/390 topology manager feeds the RODM data base with the achieved information.
6. The NMC workstation on the operator's request can retrieve the RODM data to produce the defined views.
7. Also, at initialization time, the automation managers get the order to inform the related SA OS/390 topology agents whenever status changes occur. Then the SA OS/390 topology agents will route the status change information to the SA OS/390 topology manager which will update the RODM data base.

Planning to Install the NMC Workstation

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

For information on how to install the NMC, refer to *Tivoli NetView for OS/390 Installation: Configuring Graphical Components* and *NetView Management Console User's Guide*. The information of what to do to enable your NMC environment installation for use in SA OS/390 is described in "Installing the NMC Workstation" on page 129.

Running Multiple NetViews

If you use two NetViews and you want to monitor resources using the NMC workstation, bear in mind that the NMC workstation must be linked to NetView Graphic Monitor Facility Host Subsystem (GMFHS) on the Networking NetView which has a connection to RODM. See Figure 11 on page 44. You can operate network and SA OS/390 resources via RODM and have SA OS/390 running in another NetView to control the automation resources. This, however, requires a subset of SA OS/390, referred to as the SA OS/390 satellite, to be installed on the Networking NetView. See "Step 25: Install an SA OS/390 Satellite" on page 115 for details.

If you run the Networking Automation NetView only on the focal point, then you cannot have your resources automated by SA OS/390.

If you run the System Automation NetView only on the focal point, you cannot have networking resources in RODM, but only SA OS/390 resources that you automate.

Alternatively, you can run both the Networking Automation and the System Automation on the same NetView. This way, you can save storage and CPU costs because of the reduction in the duplication of, for example, tasks and logs. But more important, it reduces maintenance and system programmer costs. See Figure 10 on page 43 for details.

In such an environment all functions are handled by that NetView. You may want to give the individual NetView tasks different priorities, for example, the System Automation tasks need to run above the VTAM's priority, whereas others (Networking Automation) need to run at a lower priority. This is achieved with OS/390 Workload Manager Enclaves support.

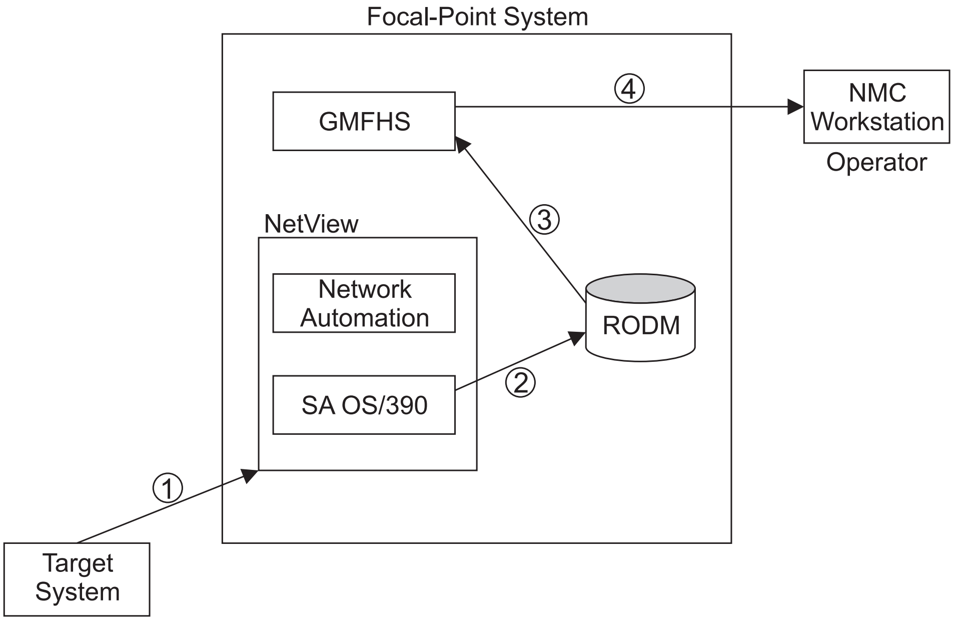


Figure 10. SA OS/390 Enterprise with Networking Automation and System Automation running on the same NetView

Figure 11 on page 44 illustrates the flow of an alert when two NetViews are used on the focal point - one for the Networking Automation and one for System Automation.

1. The alerts from the target system are sent to the Networking NetView at the focal point. (Alerts from processor operations are sent directly to the Automation NetView).
2. The satellite OS/390 automation traps the alerts and updates objects in RODM appropriately.
3. NetView Graphic Monitor Facility Host Subsystem (GMFHS) becomes aware of status updates.
4. GMFHS broadcasts updates to the operator workstation.

When an operator initiates a command or routine from a workstation, the action flows back to the Networking NetView for processing in the reverse direction from that shown in Figure 11 on page 44.

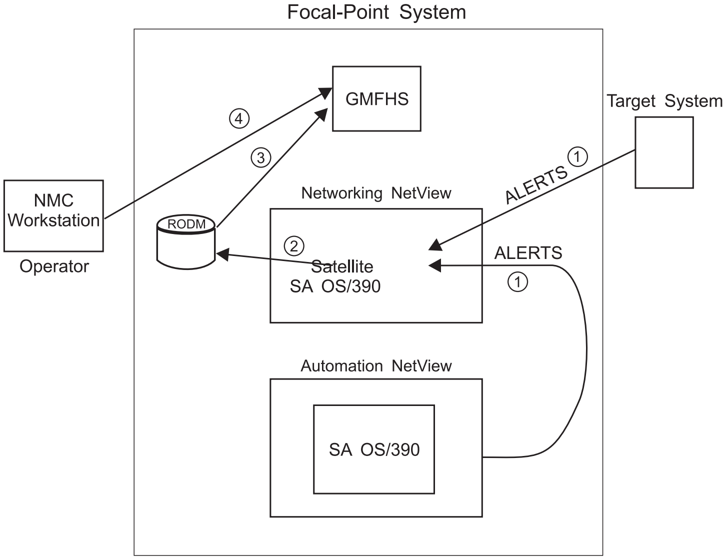


Figure 11. SA OS/390 Enterprise Using a Networking NetView and an Automation NetView

Chapter 6. Planning for Automation Connectivity

This chapter provides background on SA OS/390. It includes what a focal point system is and what targets are, and how to define a network of interconnected systems, known as an *automation network*, to SA OS/390 for purposes of monitoring and controlling the systems. The procedures and examples in this chapter assume that VTAM definitions for systems in the automation network are in place and available as input.

The Focal Point System and Its Target Systems

SA OS/390 allows you to centralize the customization, monitoring, and control functions of the multiple systems or images that make up your enterprise using a single, centrally located OS/390 system. This controlling OS/390 system is called the focal point system. The systems it controls are called target systems. These systems communicate using LU6.2 sessions, XCF, MQSeries and NetView facilities.

Defining System Operations Connectivity

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

Multiple NetViews

The number of NetViews that run in your SA OS/390 complex affects how you plan for it. SA OS/390 can operate with just one NetView at its focal point. Prior to NetView 1.4 and SA OS/390 2.2 the networking and automation function could not run on the same NetView. This is no longer the case. It is your decision whether or not you want to run the *Networking Automation* and the *System Automation* on separate NetViews.

Overview of Paths and Sessions

To learn about message forwarding paths, see “Message Forwarding Path” on page 46 and to learn about gateway sessions, see “Gateway Sessions” on page 46.

Message Forwarding Path

SA OS/390 generates and uses messages about significant actions that it detects or takes such as a resource status change. In addition to sending these messages to operators on the same system, SA OS/390 can forward them from target systems to a focal point system and can route commands and responses between systems, using a message forwarding path. This path is defined in your policy. Key components in a message forwarding path include:

- A primary focal point system
- A backup focal point system
- A target system or systems
- Gateway sessions connecting systems. Gateway sessions use inbound and outbound gateway autotasks or INGSEND to route commands and responses between systems, instead of requiring individual NetView-to-NetView task (NNT) sessions for each operator. Operators can use gateway sessions for command and response routing in an automation network. An optional feature manages the passwords used for logging the gateway autotasks on to other systems.
- One or more adjacent NetView systems that act as a path for forwarding messages, commands, and responses between systems that do not have a direct gateway connection.

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

Appendix G, “Example Automation Network Configuration”, on page 195 describes an example message forwarding path.

SA OS/390 uses notification messages to update the status of resources displayed on the status display facility (SDF). Routing notification messages over the message forwarding path helps consolidate monitoring operations for multiple systems on the SDF at a focal point system. See *System Automation for OS/390 User’s Guide* for details on configuring SDF for a focal point system-target system configuration.

Gateway Sessions

To forward messages and route commands and responses between systems, you can use either dedicated NNT sessions for each logged-on operator, or autotasks called gateway autotasks, or INGSEND. Using gateway autotasks is the method described in this chapter.

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

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

In the automation policy for each system in an automation network, you need to define only the outbound gateway autotask (see *System Automation for OS/390*

Defining Automation Policy). However, in the NetView DSIPARM data set member DSIOPF, you must define all gateway autotasks, both inbound to and outbound from a system, as operators.

You define the outbound gateway autotask by defining the GATOPER policy item for the Auto Operators policy object in the customization dialog. You must specify an operator ID associated with the GATOPER function in the Primary field on the Automation Operator NetView panel. See *System Automation for OS/390 Defining Automation Policy* for more information.

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

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

O Outbound gateway autotask
I Inbound gateway autotask.

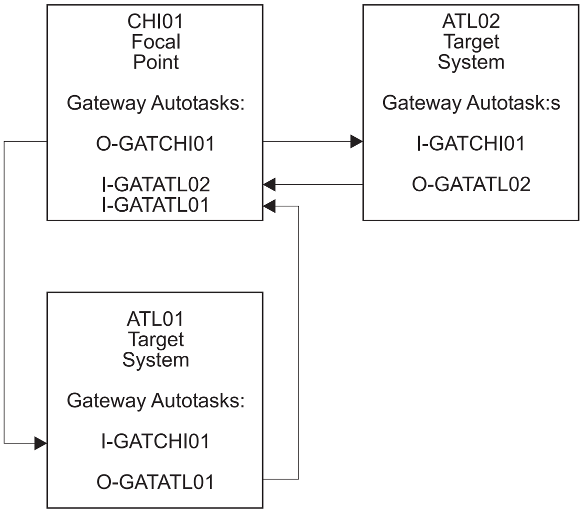


Figure 12. Example Gateways

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

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

When this message is received, the following steps occur:

1. The outbound gateway autotask tries logging on to all systems specified in gateway definitions in the automation policy for the system, using the operator ID defined for the outbound gateway autotask.
2. Once the outbound gateway autotask is logged on to the other systems, this autotask sends a message to the outbound gateway autotask on each system to which it is connected, requesting that their outbound gateway autotasks log on.
3. A gateway session between two systems is established when the outbound gateway autotasks for each system log on to the other system.

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

For example, in the gateway examples shown in Figure 12 on page 47, if message DSI112I is received on system CHI01, the outbound gateway autotask for CHI01 attempts to log on to systems ATL01 and ATL02, because they are defined as gateway entries on system CHI01. The outbound gateway autotask logs on using operator ID GATCHI01. Once the CHI01 outbound gateway autotask logs on to systems ATL01 and ATL02, as the inbound gateway autotask, it sends messages to the outbound gateway autotasks on ATL01 and ATL02, asking them to log on to CHI01. ATL01 then uses operator ID GATATL01 to log on to CHI01, and ATL02 uses operator ID GATATL02. System CHI01 then has three gateway autotasks active:

- One outbound, using operator ID GATCHI01
- One inbound from ATL01, using operator ID GATATL01
- One inbound from ATL02, using operator ID GATATL02

On each system, the Gateway Display panel of the SA OS/390 NetView interface displays the status of gateways. Refer to *System Automation for OS/390 User's Guide* for details.

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

See *System Automation for OS/390 Defining Automation Policy* for details. The ID of the timer created to monitor gateway sessions is AOFGATE. This timer will not be set if NONE is entered for Gateway Monitor Time.

If SA OS/390 detects that any gateway session is inactive during the monitoring cycle, it tries to restart the session by issuing a start domain command.

Issuing Commands Using Gateway Services: Once gateway sessions between automation agents running on different systems are established, operators can route commands to other systems using the outbound gateway autotask. Command responses can be routed from the other systems back to the originating system using inbound gateway autotasks.

Using the outbound and inbound gateway autotasks for routing commands and responses eliminates requiring all operators to have personal NetView-NetView (NNT) sessions for all systems to which they need access.

Commands can be issued using the INGSEND command. (Refer to *System Automation for OS/390 User's Guide* and *System Automation for OS/390 Operator's Commands* for descriptions of the INGSEND command.)

Adjacent NetView Paths

Your automation network may require routing messages, commands, or responses between systems that are not directly connected by a gateway session. In such cases, you need to define to SA OS/390 which existing gateway session between systems is used to send the information on to the destination system. This definition is called an adjacent NetView path description.

Automatically Initiated Terminal Access Facility (TAF) Full-Screen Sessions

Using the FULL SESSIONS policy item of the Network policy object, you can set up automatically initiated terminal access facility (TAF) full-screen sessions from within SA OS/390. *System Automation for OS/390 Defining Automation Policy* describes how to define applications with which SA OS/390 operators can establish TAF sessions automatically using the SA OS/390 NetView interface.

Automatically Initiated User NetView-to-NetView (NNT) Sessions

Using the NNT Sessions policy object on the customization dialog panels, you can set up automatically initiated user NNT sessions from within SA OS/390.

Using Focal Point Services

Once an automation network is configured, you can use the message forwarding path to route messages, commands, and responses between systems. SA OS/390 operators can display the status of gateway autotasks, NNT sessions, and TAF full-screen sessions using the SA OS/390 operator commands. Details on these operator activities are in *System Automation for OS/390 User's Guide*.

Defining Processor Operations Communications Links

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

- NVC
- SNMP

Meeting Availability Requirements

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

- Backup Support Element
- Alternate focal point system

Backup Support Element

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

Alternate Focal Point System

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

Alternate Focal Point for SNA based NVC connections

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

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

Alternate Focal Point for SNMP connections

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

BCP Internal Interface considerations

If you have customized SA OS/390 to use the BCP Internal Interface for the sysplex hardware automation, each system being a member of the sysplex has its processor hardware connection activated and can issue hardware requests to the SEs of the other sysplex members. The SA OS/390 internal code routes the supported hardware commands only to a system in the sysplex with a functioning hardware interface to make sure the request can be processed successfully.

Task Structure for Processor Operations

For processor operations there is a task structure that is modular; distinct types of SA OS/390 tasks handle different work assignments. The types of SA OS/390 tasks are:

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

SA OS/390 allows up to 999 tasks of each of the first three types, but only one recovery task and one processor operations start task. Because SA OS/390 tasks are OS/390 tasks that require system services and also add to the load running in the NetView address space, you should only define as many tasks as are needed.

The following guidelines help you match the number of SA OS/390 tasks to your SA OS/390 configuration.

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

Target Control Tasks

The number of message monitor tasks and the number of target control tasks are specified when you customize your Enterprise policy.

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

Message Monitor Tasks

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

Message monitor tasks receive SNMP traps at the focal point system from the Support Element's SNMP clients and broadcast the traps as or alerts to appropriate tasks and operators. Each message monitor task is a NetView autotask.

Recovery, Start, and Polling Tasks

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

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

The polling task, another NetView task, is used to poll the processors using NetView connections. You determine both the polling frequency and polling retries to be attempted. (These polling functions are specified using the NetView connection path definition panels in the configuration dialogs.) This task is generated automatically when you generate the NetView Autotask definitions from the customization dialogs. This NetView task enables SA OS/390 to verify and update operations command facility-based processor status.

Processor Operations OCF-CI Task

The OCF-CI task receives messages sent to the support element on the console integration interface. These messages come from a target OCF-based or parallel enterprise server operating system. The task broadcasts these messages to the appropriate processor operations task and operators interested in processor operations. For information about interested operators, see *System Automation for OS/390 User's Guide*. The OCF-CI task, a NetView autotask, is singular. Only one is required, and it is required only for processors connected with a NetView connection. This task is not required for SNMP connections.

Planning to Connect the Hardware

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

- "Preparing the Focal Point System Connections" and "Alternate Focal Point System Considerations" on page 52 for focal point system connections
- "Preparing the Target System Connections" on page 53 for target system connections. This section also discusses complex connection configurations.

Preparing the Focal Point System Connections

The physical path for the focal point system consists of connections from the PC to the focal point system. SA OS/390 processor operations supports the following types of communication connections:

- NVC
- SNMP

Alternate Focal Point System Considerations

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

Applications

An alternate focal point system allows you to:

- Monitor and control your DP enterprise from another physical location. By installing an alternate focal point system at this location, you can switch from the primary focal point system to the alternate focal point system and back again.
- Provide a method for minimizing the effects of a primary focal point system outage. If a focal point system must remain operational at all times, an alternate focal point system with a separate communications network acts as a stand-by focal point system. Backup personal computers at the target systems can provide further backup protection.

Preparing the Alternate Focal Point System Connections

The physical connection path for the alternate focal point system is identical to that for the primary focal point system. As with the primary focal point system, SA OS/390 processor operations supports the following types of communication connections:

- NVC
- SNMP

The graphic in Figure 13 on page 53 illustrates how primary and alternate processor operations focal point systems are connected to a target processor.

Connection Example: Figure 13 on page 53 shows an alternate focal point system as well as a primary focal point system connected from an IP or SNA network to the processor hardware LAN.

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

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

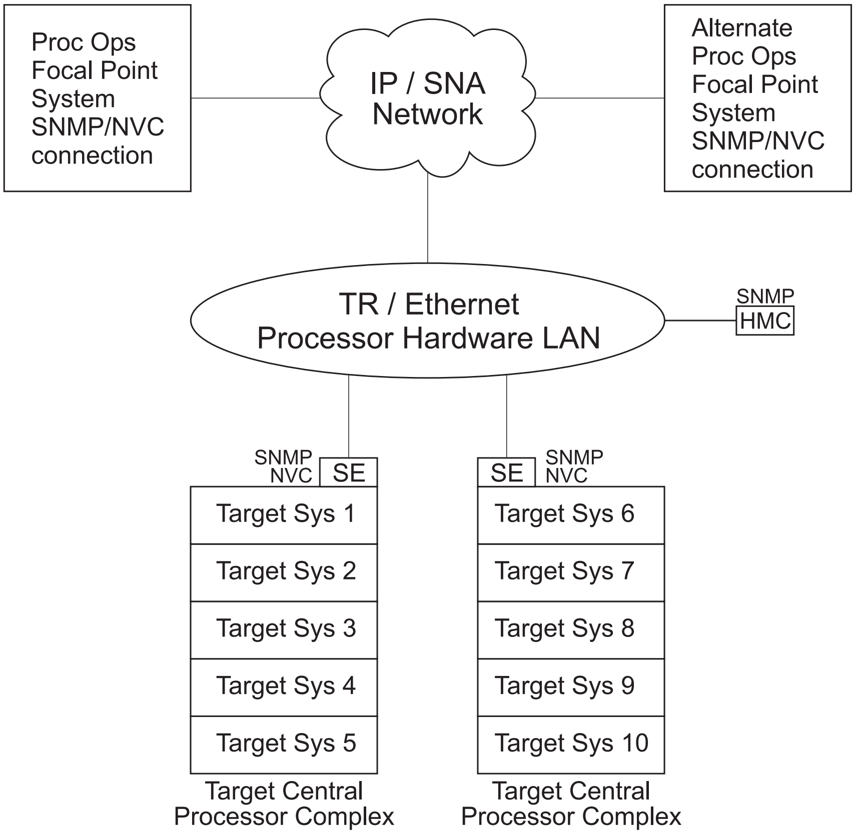


Figure 13. Alternate and Primary Focal Point System Connections from an IP or SNA Network to the Processor Hardware LAN

Preparing the Target System Connections

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

The usage of CI by processor operations is intended to automate system initialization and recovery tasks. For day-to-day console operation tasks, processor operations CI usage should supplement the operating system command routing facilities of SA OS/390 or the available console devices like the 2074 control units.

Defining I/O Operations Communications Links

When you use SA OS/390 on one system to make an operational change to an I/O resource, like a shared ESCON Director, it coordinates the change with other copies of SA OS/390 on other systems. This is especially important when the result of the action you are taking removes connectivity - disables I/O paths - so that the systems do not lose access to critical resources. Each copy of SA OS/390 interacts with its local system image (for example, via VARY) so the operating system has the chance to ″vote″ on the changes. When one system fails VARYs, SA OS/390 takes that as a vote of ″no″ and fails the operation. The copy of SA OS/390 from which you initiated the operation then interacts with the other copies on the affected system images to back out VARYs that were successful.

The copies of SA OS/390 across your systems also use the network to share information with each other on changes to the I/O configuration and to provide displays that collect I/O information from multiple systems.

To do this, the SA OS/390 I/O operations functions on each system image need to intercommunicate. They do this by establishing VTAM sessions between each other. All systems that share access to a given ESCON Director should run SA OS/390 to provide the protection described above. Those copies of SA OS/390 that do share access to a Director automatically discover each other and establish sessions each time they start.

You can also use the Reset Host function of I/O operations to force two copies of SA OS/390 that do not share any ESCON Directors to establish communications. This is useful if you want to benefit from the I/O operations multisystem I/O graphic displays or use its multisystem version of Remove CHP, Restore CHP, Remove Device, or Restore Device, even across system images that don’t use ESCON Directors or have no reason to share them.

I/O operations is able to interact with systems that are running ESCON Manager. I/O operations can interact with VM systems that run ESCON Manager 1.2 to support switching operations (for example, blocking ports or writing an entire saved switch configuration) and for Remove CHP and Restore CHP. I/O operations can interact with OS/390 systems that run ESCON Manager 1.3 to support the same operations as for VM, and also the same level of multisystem Query and graphic display requests that ESCON Manager 1.3 itself supports.

To plan for this function, you must review the I/O configuration across the systems that you will define as an enterprise in SA OS/390. You should plan to include in one enterprise all system images that share a given ESCON Director, in order to benefit from the I/O operations configuration change protection and displays.

To enable the VTAM sessions, you must create VTAM definitions as described in “Step 17B: Perform VTAM Definitions” on page 107 to support communications between I/O operations defined as a VTAM application in each of them.

Where images do not automatically use those definitions to start sessions, because they do not share ESCON Directors, you should plan local procedures to use the SA OS/390 Reset Host function to force I/O operations to start the sessions.

Chapter 7. Naming Conventions

SA OS/390 System Names

The information in this section describes name requirements for OS/390 systems and for processor operations functions.

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

If your system names currently contradict this restriction, you must change the names before using SA OS/390.

System names defined in the customization dialog for OS/390, VM, TPF, or LINUX systems can have up to 20 characters and must be unique within the SA OS/390 enterprise.

When you name elements of your SA OS/390 processor operations, use a logical format to create names that are clear to the people using them. The following names can consist of 1 to 8 alphanumeric characters (A-Z,a-z,0-9,#,$,@), cannot contain blanks, and must begin with an alphabetic character:

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

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

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

Cloning on OS/390 Systems

The SA OS/390 cloning capability allows you to specify up to 36 clone IDs to identify a system and to identify an application. These clone IDs are then used to qualify the application job name to ensure a unique job name for each system. The names given to each of these clones must be unique. The OS/390 system symbolics and the NetView &domain. variable can also be used.

Further Processor Operations Names

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

ESCON Director Ports

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

Reasons for Naming Switch Ports

Assigning names to switch ports:

- Provides an indication of what is on that port. For example, CP01.SYSA.CHP38 indicates that this port is physically connected to processor CP01, on system SYSA, on CHPID 38.
- Allows you, when issuing I/O operations connectivity commands, to refer to ports by name. For example, `BLOCK 3490.46233.CU1.E *` blocks the port connected to interface E of control unit side 01, on the 3490 control unit with serial number 46233. See “Using Port Logical Names” on page 57.
- Allows you, when issuing I/O operations connectivity commands, to change connectivity of an entire system to a control unit. For example, `PROHIBIT CP01.SYSA* 3990.35182* *` removes connectivity from all ports on system SYSA of processor CP01, from all ports on the 3990 control unit with serial number 35182. See “Using Generic Logical Names” on page 58.

Suggestions for Naming ESCON Director Ports

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

Naming CHPID Ports

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

```
CP02.SYSC.CHP40
```

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

Naming Control Unit Ports

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

```
3990.35182.SC1.E
```

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

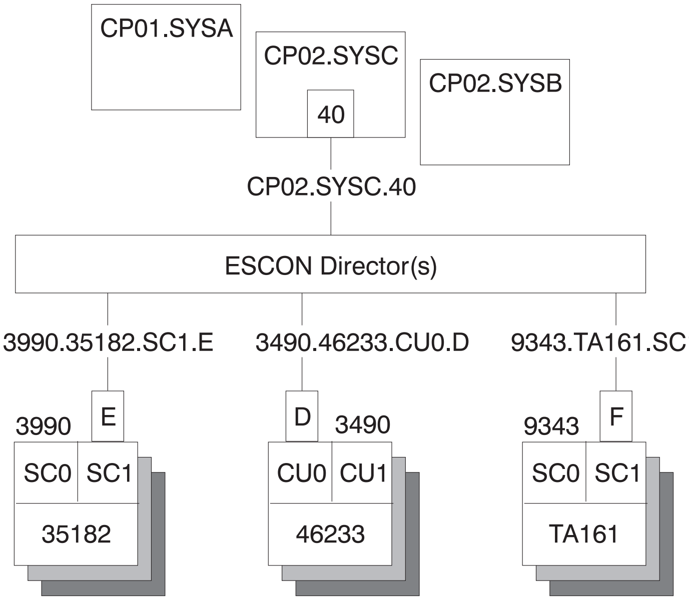


Figure 14. Examples of Port Names in a Configuration

Methods of Naming Ports

You can assign names to ports using the following:

- The WRITE command

  You can use the command:

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

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

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

  You can create an EXEC with commands like:

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

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

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

Using Port Logical Names

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

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

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

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

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

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

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

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

Using Generic Logical Names

SA OS/390 I/O operations provides the ability to use an asterisk as a wild card character in commands that use port names. This allows you to make more than one change on each switch.

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

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

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

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

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

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

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

Command Usage Examples with Generic Logical Names

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

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

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

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

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

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

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

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

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

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

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

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

Part 2. Installation

Chapter 8. Installing SA OS/390 on Host Systems

This chapter describes the tasks required to install SA OS/390 components on the SA OS/390 host systems. This chapter includes information on installing SA OS/390 on both focal point and target systems. The target system installation does not require some of the steps used for the focal point installation. Any installation step that does not apply to the target systems is indicated. Many of the installation steps have corresponding planning activities and explanations in chapters 2 through 6 of this book. Chapter 9 describes installation on workstations.

In this chapter, the single installation steps are marked as either being required for all or certain SA OS/390 components or as being ***optional***. ***Optional*** denotes steps that may or may not need to be performed based on your environment, your system management procedures, and your use of the SA OS/390 product. For each of these steps you need to decide whether it is required for your installation.

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

Notes:

1. The meaning of the term *target system* as used by SMP/E needs to be distinguished from the way the term is used in SA OS/390. As used in SMP/E and when describing the installation of OS/390 products and services, a target system is the system on which a product such as SA OS/390 is installed. It is the collection of program libraries that are updated during SMP/E APPLY and RESTORE processing. In this publication this meaning of target system is referred to as an “SMP/E target system”. The usual SA OS/390 meaning of a “target system” is a computer system attached to a focal point system for purposes of monitoring and control.
2. In this book, data set names are shown with the high level qualifier ING. You can have a different high level qualifier for your data sets.
3. If ESCON Manager is already installed, consider that SA OS/390 ***cannot*** run together with ESCON Manager on the same system. Running a mixed environment will end up with unpredictable results for example, storage overlay ABEND0C4 or ABEND0C1. See also “Step 4D: Update LPALST*xx*” on page 72 and “Step 4E: Update LNKLST*xx*” on page 72.

Overview of Installation Tasks

The major tasks required for installing SA OS/390 on a focal point are listed in Table 10.

Table 10. Installation Tasks for SA OS/390 Host Systems

Description	Page	Required for	Optional for
Step 1: SMP/E Installation	66	SysOps, ProcOps, I/O Ops	
Step 2: Allocate Data Sets Common to Systems	67	SysOps, ProcOps, I/O Ops	
Step 3: Allocate Data Sets for the Customization Dialog	68	SysOps, ProcOps, I/O Ops	
Step 4: Customize SYS1.PARMLIB Members	70	SysOps, ProcOps, I/O Ops	
Step 5: Setting up MQSeries	74	SysOps	
Step 6: Customize SYS1.PROCLIB Members	77	SysOps, ProcOps, I/O Ops	
Step 7: Customize NetView	78	SysOps, ProcOps	
Step 8: Preparing the Hardware	84	SysOps, ProcOps	
Step 9: Customizing the Automation Manager	89	SysOps	
Step 10: Customizing the Component Trace	91	SysOps, ProcOps	
Step 11: Customizing the System Logger	91		SysOps
Step 12: Install ISPF Dialog Panels	93	SysOps, ProcOps, I/O Ops	
Step 13: Verify the Number of available REXX Environments	101	SysOps, ProcOps	
Step 14: Customization of NetView for TEC Notification by SA OS/390	102		SysOps, ProcOps
Step 15: Compile SA OS/390 REXX Procedures	105		SysOps, ProcOps
Step 16: Defining Automation Policy	105	SysOps, ProcOps	
Step 17: Define Host-to-Host Communications	106	SysOps, ProcOps, I/O Ops	
Step 19: Enabling SA OS/390 to Restart Automatic Restart Manager Enabled Subsystems	110		SysOps
Step 18: Add Additional VTAM APPL Statements	110	SysOps, ProcOps	
Step 20: Define Security	110		SysOps
Step 21: Customize the Status Display Facility (SDF)	111		SysOps
Step 22: Check for Required IPL	112	SysOps, ProcOps I/O Ops	
Step 23: Automate System Operations Startup	112	SysOps	

Table 10. Installation Tasks for SA OS/390 Host Systems (continued)

Description	Page	Required for	Optional for
Step 24: Verify Automatic System Operations Startup	114		SysOps
Step 25: Install an SA OS/390 Satellite	115		SysOps
Step 26: Installing and Customizing the NMC Focal Point	116		SysOps
Step 27: Copy and Update Sample Exits	123		SysOps, ProcOps I/O Ops

Step 1: SMP/E Installation

Required for:

SysOps, ProcOps, I/O Ops

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

Note: In the steps that follow, sample jobs are all members of the SINGSAMP data set, the SA OS/390 sample library.

Table 11 shows a list of target data sets as provided by the SMP/E installation process to be used for production on your system.

Table 11. Target Data Sets

Data Set Name	Description
ING.SINGIMOD	SA OS/390 modules for ISPF **1**
ING.SINGIMSG	ISPF messages **1**
ING.SINGINST	SMP/E jobs to install the product alternatively to using SMP/E dialogs **2**
ING.SINGIPDB	policy database samples **1**
ING.SINGIPNL	ISPF panels **1**
ING.SINGIREX	ISPF REXX execs **1**
ING.SINGISKL	ISPF skeletons **1**
ING.SINGITBL	ISPF tables **1**
ING.SINGMOD1	different SA OS/390 modules **3**
ING.SINGMOD2	different SA OS/390 modules in LINKLST **3**
ING.SINGMOD3	different SA OS/390 modules in LPALIB **3**
ING.SINGNMSG	NetView messages **3**
ING.SINGNPNL	NetView panels **3**
ING.SINGNPRF	NetView profiles **3**
ING.SINGNPRM	NetView DSIPARM samples **3**
ING.SINGNREX	NetView REXX execs **3**

Table 11. Target Data Sets (continued)

Data Set Name	Description
ING.SINGSRC	SA OS/390 source 3
ING.SINGPWS1	NMC exploitation code 4
ING.SINGSAMP	General samples 3
ING.SINGMODC	for CICS 5
ING.SINGMODI	for IMS 6

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

1 Data sets of this category are related to ISPF and need to be accessed by all persons using the customization dialog.

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

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

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

5 Data sets of this category need to be accessed by system programmers responsible for CICS automation.

6 Data sets of this category need to be accessed by system programmers responsible for IMS automation.

Step 2: Allocate Data Sets Common to Systems

Required for:

SysOps, ProcOps, I/O Ops

The sample jobs in INGALLC1, INGALLC2, INGALLC3, and INGALLC4 in SINGSAMP can be used to allocate data sets required for both focal point and target systems. In the comments included in the sample job are instructions for customizing the values for the data set allocations. These data sets are:

- ***for I/O operations***

Table 12. Data Sets for I/O Operations

Data Set	DD Name	Sample
HCD trace file	HCDTRACE	INGALLC1

- ***for all automation agents***

Step 2: Allocate Data Sets Common to Systems

These data sets are required once per automation agent and cannot be shared.

Table 13. Data Sets for Automation Agents

Data Set	DD Name	Sample
automation status file	AOFSTAT (in the NetView start procedure)	INGALLC2
dump file	INGDUMP	INGALLC2
gateway password file	AOFPSWD (in the NetView start procedure)	INGALLC2
IPL data collection	HSAIPL (in the NetView start procedure)	INGALLC4

- ***for all automation managers***

  The following data sets are required once per sysplex or stand-alone system.

Table 14. Data Sets for the Sysplex or Stand-Alone System

Data Set	DD Name	Sample
schedule override file	HSAOVR	INGALLC3
configuration information data set	HSACFGIN	INGALLC3
component trace data set	n/a	INGALLC3
PARMLIB	HSAPLIB	INGALLC3
takeover file (1)	HSATKOVR	INGALLC3
Notes: 1. Calculate the required size of the takeover file, which is a VSAM data set, by the following formula: 4000 records + *n* records of 4K where *n* is the maximum numbers of resources.		

The following data sets must be allocated once for each automation manager. For more details, see the comments in the INGALLC3 sample.

Table 15. Data Sets for the Automation Manager

Data Set	DD Name	Sample
internal trace files (optional)	TRACET0, TRACET1	INGALLC3
SYSOUT data set	SYSOUT	INGALLC3

Step 3: Allocate Data Sets for the Customization Dialog

Required for:

SysOps, ProcOps, I/O Ops

When this step is performed, the sample job INGEDLGA in SINGSAMP can be used to allocate data sets required for the I/O operations and the customization dialog. These data sets are normally allocated only on the focal point system, where you use the customization dialog. These data sets include:

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

  The following data sets are created:

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

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

  The following data sets are created:

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

- *for I/O operations*
  - The I/O operations configuration file. Since you use the customization dialog to collect information and build control files, you normally need them only at the focal point. The I/O operations dialogs, however, are used to input commands and get responses from the I/O operations part of SA OS/390. Since they do not support multisystem commands for I/O operations functions, you must install them on each system, focal point or target, where you wish to use them.

  The following data sets are created:

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

Notes:

1. Keep these data set names in mind. They are used in “Step 12: Install ISPF Dialog Panels” on page 93. If you rename the data sets, you need to adapt the corresponding names in that step.
2. You might also allocate a dedicated data set for the output for building the system operations control files (automation control file and automation manager configuration file). The default for this data set is the policy database itself. It is however recommended to use a separate data set.

Step 4: Customize SYS1.PARMLIB Members

Required for:

SysOps, ProcOps, I/O Ops

The *xx* suffix on each SYS1.PARMLIB data set member can be any two characters chosen to match your IEASYS naming scheme. The SA OS/390 samples delivered in SINGSAMP use a suffix of *SO*. See *z/OS MVS Initialization and Tuning Reference* for information about IEASYS.

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

Step 4A: Update PROG*xx*

Required for:

SysOps, ProcOps, I/O Ops

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

Update PROG*xx* to include:

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

  Note: Do not include SYS1.NUCLEUS.
- If you chose to set AOF_SET_AVM_RESTART_EXIT to 0 in module AOFEXDEF then you will need to add this entry into your PROG*xx* member:

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

Depending on the version of IMS you are executing, authorize:

ING.SEVIMOD6 for Version 6 of IMS.

ING.SEVIMOD7 for Version 7 of IMS.

ING.SEVIMOD8 for Version 8 of IMS.

Note: Do this authorization only if the ING.SEVIMOD* library is going to be added to the STEPLIB chain of the control region JCL.

Step 4B: Update SCHED*xx*

Required for:

SysOps and ProcOps

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

This enables the NetView subsystem interface job, the NetView application job (for the automation agent) and the automation manager to run without being swapped out of memory.

Step 4C: Update MPFLST*xx*

Required for:

SysOps and ProcOps

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

This adds the SA OS/390 message automation and console display suppression specifications to the MPFLST*xx* member.

The *AUTO(YES)* in the NO_ENTRY statement is required to gather all unknown WTORs. If you ensure that the unknown WTORs are routed to automation via the general MPF exit IEAVMXIT and you have all messages that are specified in the NetView message automation table also specified in the MPF with *AUTO(YES)*, you can specify *AUTO(NO)* for the NO_ENTRY statement.

If you want to use CICS Automation, add the following entries in MPFLST*xx* in SYS1.PARMLIB to trap all DFH , EVE, and EYU prefix messages:

```
DFH*,SUP(NO),AUTO(YES)
EVE*,SUP(NO),AUTO(YES)
EYU*,SUP(NO),AUTO(YES)
```

This CICS Automation requirement forwards these messages to NetView.

Add the following entry in MPFLST*xx* in SYS1.PARMLIB to trap all DFS and EVI prefix messages:

```
DFS*,SUP(NO),AUTO(YES)
DSP*,SUP(NO),AUTO(YES)
EVI*,SUP(NO),AUTO(YES)
IOS071*,SUP(NO),AUTO(YES)
DXR*,SUP(NO),AUTO(YES)
AVM0*,SUP(NO),AUTO(YES)
IEF450*,SUP(NO),AUTO(YES)
```

This IMS Automation requirement forwards these messages to NetView.

Step 4: Customize SYS1.PARMLIB Members

Add the following message entry in MPFLST*xx* in SYS1.PARMLIB to trap all Tivoli OPC and Tivoli Workload Scheduler for z/OS messages:

```
EQQ*,SUP(NO),AUTO(YES)
EVJ*,SUP(NO),AUTO(YES)
```

This OPC/TWS Automation requirement forwards these messages to NetView.

See *z/OS MVS Initialization and Tuning Reference* for more information about MPF.

Messages processed by the automation either via the NetView MAT (message automation table) or by the NetView commands TRAP and WAIT must not be suppressed by any MPF (message processing facility) list being used.

The following messages must be available for the Parallel Sysplex automation:

```
IEA230E IEA231A IEA232A IEA404A IEA405E IEA406I IEA794I
IEE037D IEE041I IEE043I IEE205I IEE400I IEE503I IEE533E IEE600I IEE712I IEE769E IEE889I
ILR009E
INGY1097I
IRA200E IRA201E IRA202I IRA204E
IXC102A IXC247D IXC250I IXC251I IXC255I IXC402D IXC500I IXC501A IXC517I IXC559I
IXC560A
IXG257I IXG261E
IXL126I IXL127A
```

Step 4D: Update LPALST*xx*

Required for:

SysOps, ProcOps, I/O Ops

Edit the LPALST*xx* member to add ING.SINGMOD3 to the SA OS/390 load library. There is no other choice for this library, it must be in the LPALST concatenation.

You can avoid an IPL:

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

Notes:

1. Make sure that the SA OS/390 load library is cataloged in the master catalog or copy the members in ING.SINGMOD3 into a data set which is in the master catalog.
2. Be sure you do not have any data sets containing load modules with prefixes of **IHV, AOF, ISQ, ING, HSA** in these members.

Step 4E: Update LNKLST*xx*

Required for:

SysOps, ProcOps, I/O Ops

To run SA OS/390, you must ensure that program libraries can be found at start up time.

Add *SINGMOD2* to the LNKLST concatenation. There is no other choice for this library: it **must** be in the LNKLST concatenation.

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

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

OS/390 link list data sets no longer have to be cataloged in the master catalog. It is possible to specify a volume in the link list entry for data sets which are cataloged in user catalogs.

Note: Be sure you do not have any data sets containing load modules with prefixes of **IHV, AOF, ISQ, ING, HSA, EVE, EVI, EVJ** in these members.

Add the following data sets:
- ING.SINGMOD1, ING.SINGMOD2
- SYS1.SCBDHENU (for I/O operations)

Step 4F: Update IEFSSN*xx*

Required for:

SysOps and ProcOps

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

Compare the contents of the IEFSSN*xx* member with the INGESSN member, which resides in the SA OS/390 sample library.Edit the IEFSSN*xx* member so that it includes the subsystem records from the INGESSN member.

This defines:
- Four-character prefix used in the NetView startup procedure member names. The four-character prefix that you specify must match the four-character prefix of the NetView startup procedure member names. For example, if you specify INGE, then the names of the NetView startup procedure members must be INGE*xxxx*, where *xxxx* are any four characters you choose. If you change this four-character prefix, you can dynamically add this entry using the OS/390 command SETSSI. Otherwise you must perform an IPL of OS/390 to effect the change.
- JES startup specifying JES2 or JES3 with the NOSTART option. This prevents JES from starting before SA OS/390 during the IPL process. If you plan to start JES before NetView, remove the NOSTART option from the following statement:

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

  You can also use the IEFSSN-syntax:

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

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

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

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

- Make sure that the SA OS/390 SSI entry is the first SSI in the SSN table and the SSI starts during the IPL.
- Use an option available with NetView to specify “NOPPI” on all NetView SSIs except the SSI that product automation uses. This “NOPPI” option is specified as a startup parameter on the SSI JCL.
- If you do not code the SSI that product automation uses in the highest position in the SSN table and you do not use the “NOPPI” option, then the SSI that is first in the SSN table must be up before product automation initialization and must remain uninterrupted until final termination of product automation.

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

Step 4G: Update JES3IN*xx*

Required for:

SysOps

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

This includes the DUMP options and adds the JES3 parameters.

Step 5: Setting up MQSeries

Optional for:

SysOps

When you want to use MQSeries for communication between the automation manager and the automation agents and provide a continuous high-reliable environment for the automation manager, you must set up an MQSeries manager. The basic steps on how to do this are described in *MQSeries for OS/390 System Management Guide*. The outline for setting up MQSeries for exploitation by SA OS/390 is described in the subsequent substeps.

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

Step 5A: Customizing an MQSeries Manager for SA OS/390

This needs to be done on every system where either the automation manager and/or an automation agent is installed.

Please refer to *MQSeries for OS/390 System Management Guide* for the correct MQSeries installation and setup processing. It is recommended that there is a single MQSeries manager instance for SA OS/390. The way SA OS/390 exploits the MQSeries infrastructure does not immediately require a dedicated DB2 for MQSeries's shared data repository.

The following list describes which parameters and MQSeries options need special consideration for SA OS/390.

- SA OS/390 is using the TSO/Batch adapter
- SA OS/390 does not require any distributed queuing capabilities
- Archiving can be switched off. See Macro CSQ6LOGP.
- The maximum number of connections SA OS/390 is using is in the range of 20 which is the current default. However if the number of SA OS/390 query threads is increased drastically you may also require additional connections. See Option IDBACK in Macro CSQ6SYSP.
- The maximum number of messages processed per MQSeries transaction is normally set to 10000 via CSQINP1. See MQSeries DISPLAY MAXSMSGS. This should be sufficient in all cases. However when it turns out that the number of messages in the State Queue (see INGAMS commands) reaches the area of 4000, this value should be set to approximately. 2.5 * the number of expected state queue messages. See MQSeries DEFINE MAXSMSGS.
- For the calculation of the number of log records see the *MQSeries for OS/390 System Management Guide*. You may consider that the largest processing load which is to be logged is the takeover case, where about 5000 MQPUTs and 5000 MQGETs with an average of 4 K messages are produced in a timeframe of one minute. As a rule of thumb, you can take the MAXSMSGS value, divide it by two to get the number of maximal MQGETs. The same number can be taken for the maximum number of MQPUTS. The largest transaction producing so many GETs and PUTs should be in a range of a minute and processes 4 K messages.

ARM Considerations for MQSeries Manager

If you choose OS/390 Automatic Restart Manager for doing the restart, the MQSeries manager instance must be set up to allow element restarts only. A cross system restart is not required.

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

In a full sysplex environment, you need to build CF list structures needed for MQSeries shared queues. One CF list structure can have more than one MQSeries queue, however a queue cannot span CF structure boundaries. It is recommended that you use two CF structures for the SA OS/390 queues. The Workitem Queue and the Agent Queue can easily share a CF structure. The CF storage size consumption for the Automation State Queue could be very dynamic, because the automation manager can generate a huge amount of uncommitted MQSeries messages also using CF storage.

Please refer to the *MQSeries for OS/390 System Management Guide* for information on how to calculate the size of the CF List Structures. The number of messages per queue and the message size can be taken from the provided samples (INGALLMS, INGALLML). Keep in mind that you have to double the number of messages for the state queue because of uncommitted updates which also occupy CF storage.

Step 5C: Definition of MQSeries Queues

Two sample jobs are delivered defining the queues either as

- local Queues for a single system environment with Sample INGALLML
- shared Queues for a full Sysplex Environment with Sample INGALLMS

Important operands to consider:

name of the queue
the names of the queues are fixed and follow the pattern:

- 'HSA' the SA OS/390 automation manager component prefix
- the XCF groupID (8 Characters) to allow more than one Automation domain per Sysplex

  This is the only modifiable part. Please see the samples.
- a predefined character string as suffix:
  - WORKITEM.QUEUE for the Workitem Queue
  - STATE.QUEUE for the Automation State Queue
  - AGENT.QUEUE for the Automation Agent Queue

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

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

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

Display MQSeries Statistics

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

It is also possible to use the ISPF based MQSeries standard operations and control panel to operate with the SA OS/390 queues.

Step 5D: RACF Considerations for MQSeries

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

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

Notes:

1. `mqsubsysid` is the 4 character subsystem name of the local MQSeries queue manager.
2. `qsgname` is the name of the queue-sharing group to which the queue manager belongs (refer to the QSGDATA statement in the MQSeries documentation for details on `qsgname`.
3. The SA OS/390 automation manager and automation agents must be granted READ access to the resource profiles of MQCONN.
4. The SA OS/390 automation manager and automation agents must be granted ALTER access to the resource profiles of MQQUEUE.

Step 6: Customize SYS1.PROCLIB Members

Required for:

SysOps, ProcOps, I/O Ops

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

- ***for system operations only***
  - Include the procedure that runs the automation manager into SYS1.PROCLIB

    A separate NON-APF authorized task library is required in addition to the authorized steplib.
    - The NON-APF authorized task library is used by the LE task. It must concatenate the NON-APF authorized SA OS/390 product library with the LE runtime library and the C/C++ library.
    - The steplib specifies the APF authorized SA OS/390 product library.

    You can find an example in member HSAMPROC in the SINGSAMP sample library.

    If you do not want to place the automation manager PARMLIB member into the SYS1.PARMLIB concatenation, include a HSAPLIB DD statement into the automation manager startup procedure (see also “Step 9: Customizing the Automation Manager” on page 89):

    ```
    HSAPLIB DD DSN=IBMING.PARMLIB, DISP=SHR
    ```
  - Copy the following members from the ING.SINGSAMP:

    ```
    INGPIXCU
    INGPHOM
    INGPIPLC
    HSAPIPLC
    ```

    Follow the customization instructions that are contained within these members.

    These procedures make use of certain data sets and must have the appropriate authorizations. For details refer to “Granting NetView and the STC-User Access to Data Sets” on page 153.

  - Optional: Startup procedure for the External Writer of the Component Trace

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

- ***for system operations and processor operations***
  - NetView subsystem interface startup:

    NetView provides a sample subsystem interface startup procedure in member CNMSJ010. Copy this member from your NetView library and adapt it to your needs. Rename it to agree with the four-character prefix defined in the IEFSSN*xx* member which is described in “Step 4F: Update IEFSSN*xx*” on page 73.

  - NetView application startup:

    NetView provides a sample application startup procedure in member CNMSJ009. Copy this member from your NetView library and do the following:

    1. Check the INGENETV member residing in the SINGSAMP sample library. This is not an executable procedure, but it contains statements required for the NetView application startup procedure. Include all of these statements into your copy of the NetView application startup procedure.
    2. Ensure that the edited copy of the NetView application startup procedure contains the appropriate data set concatenation order.
    3. Ensure that your new NetView application startup procedure member uses the correct data set names for your system.
    4. Rename the NetView application startup procedure member to agree with the four-character prefix defined in the IEFSSN*xx* member, which is described in “Step 4F: Update IEFSSN*xx*” on page 73. For example, if the name of the NetView application startup procedure is INGE*xx*, then INGE must be specified in the IEFSSN*xx* member as the character prefix.

       If you do not make ING01 your domain name, note carefully what your NetView domain name is. This information is needed in the next step and for system operations. See also *System Automation for OS/390 Defining Automation Policy* for more information on enterprise definitions.

       See *Tivoli NetView for OS/390 Installation: Configuring Additional Components* for more details on how to modify the NetView startup procedure.

- ***for I/O operations***
  - Copy INGEIO into SYS1.PROCLIB and edit it to specify the domain name that you noted in step 4. The DD statement in the procedure should match the allocation you did in “Step 2: Allocate Data Sets Common to Systems” on page 67 for HCDTRACE.

Step 7: Customize NetView

Required for:

SysOps and ProcOps

This section discusses how to customize several aspects of NetView:

- “Step 7A: Customize NetView Alert Information”
- “Step 7B: Customize NetView DSIPARM Data Set”
- “Step 7C: Modifying NetView DSIPARM Definitions for an Automation Network” on page 82
- “Step 7D: Customize NetView for Processor Operations” on page 83

Step 7A: Customize NetView Alert Information

Required for:

SysOps

SA OS/390 enterprise monitoring depends upon alert information being passed from remote systems to the focal point.

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

In order to ensure that the alerts required by SA OS/390 for enterprise monitoring are not filtered out, the following is recommended:

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

These **SRF** commands should be included into a startup CLIST/exit because they need to be issued after every NetView startup.

If you do not want to allow all alerts to pass by using the

```
SRF AREC PASS N *
```

command, you should allow the following event types (*etype*s) to pass, as a minimum:

- NTFY

The NetView SRFILTER command is documented in *Tivoli NetView for OS/390 Command Reference Vol. 1*.

Step 7B: Customize NetView DSIPARM Data Set

Required for:

SysOps and ProcOps

Step 7: Customize NetView

Copy any DSIPARM and SINGNPRM member you need to customize into a data set allocated in DSIPARM above the SMP/E maintained NetView DSIPARM and SA OS/390 target libraries and edit it there.

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

CNMSTYLE
To enable SA OS/390, make sure that the following TOWER statements are activated in CNMSTYLE:

```
TOWER =  SA
TOWER.SA  = license
```

DSIDMNK

If you are in a multi-system environment, planning to use communications to other NetViews, there must be a resource routing definition statement (RRD statement) defined for each NetView system to which your system is connected. Additionally, the alert forwarding definition to your NMC should be defined. `ALERTFWD NV-UNIQ` is recommended.

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

1. Review your use of all other synonyms. Note that your JES name synonym (%AOFJES3JOB%) is used for that JES subsystem that you are using. The only reference to %AOFJES3JOB% is in the AOFMJES3 member of the SINGNPRM library. This member contains automation entries for the JES3 subsystems only.
2. If you have renamed any automation tasks in `AOFOPF`, you will need to make the corresponding changes here.
3. If you do not wish to use the `ASSIGN BY JOBNAME` exploitation (that is, Advanced Automation CGLOBAL `AOF_ASSIGN_JOBNAME` has been set to 0), do NOT specify (*) in the %AOFOPGSSOPER% synonym, because this may cause serialization problems.

INGMSG01
The MAT member INGMSG01 provides message suppression that is necessary to prevent mismatches and duplicate automation before the first %INCLUDE. The fragment INGMSGU1 can be used for user entries. These entries have precedence over the SA OS/390 entries. The default INGMSGU1 is an empty member. The fragment INGMSG02 contains all entries provided by SA OS/390. It is being automatically built from an abstract source and should not be modified. The fragment INGMSGU2 can be used for all entries that SA OS/390 does not supply. The default INGMSGU2 is an empty member.

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

```
INGMSG01
    |
    +-- %INCLUDE INGMSGU1
    |
    +-- %INCLUDE INGMSG02
    |
    +-- %INCLUDE INGMSGU2
```

INGXINIT
The communication DST initialization processing will read data that is

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

GRPID
2-byte XCF group ID. Default is blank.

MQM 4-byte MQ manager name. Use this parameter only if you run MQSeries for communication. If you use XCF, you do not need to specify MQM.

All parameter values must match with the respective parameters in the PARMLIB member HSAPRM*xx* of the automation manager.

A ***GRPID*** may be specified to indicate that a subset of the members of an actual OS/390 sysplex is defined in a sysplex group. If specified, the ID may contain 1 or 2 characters. Valid characters are A-Z, 0-9, and the national characters ($, # and @).

It will be prefixed with the string INGXSG to construct the XCF group name used for cross system synchronization, for example, `INGXSGxy`.

If no ***GRPID*** is specified, the default group name INGXSG will be used.

This group needs to be associated with an appropriate XCF transport class by the person responsible for the sysplex setup (see *z/OS MVS Setting Up a Sysplex*).

> **Note:**
> Syntax errors are reported by a message with error code ERRCODE=564. Any syntax errors will stop the initialization process and therefore no automation will be possible.

The following parsing syntax applies:

- Data can only be specified via key-value-pairs.
- One or more parameters may be specified on one line.
- Each record will be parsed for the keyword.
- Parsing will be stopped and any further input data will be ignored after all keywords listed above are found.
- If the same parameter is specified multiple times, the last one is used.
- For any keyword that was not specified, the default value is blank.
- No blanks between parameters and values are allowed.
- The syntax of a keyword is equal to the syntax of the parmlib member HSAPRM*xx*.

Example of a valid syntax:

```
GRPID=XY,MQM=SSSS
```

Example of an invalid syntax:

```
GRPID = 34 , MQM = SSSS
```

DSICMSYS / AOFCMDSO
If you want to use the SA OS/390 SETTIMER command instead of the

NetView SETTIMER command, comment out the SETTIMER synonym definition (CMDSYN SETTIMER) in DSICMSYS and uncomment (activate) it in member AOFCMDSO of the DSIPARM data set.

Step 7C: Modifying NetView DSIPARM Definitions for an Automation Network

Note: The following information refers to setting up a single NetView automation network.

To support an automation network, you need to add or modify NetView definitions in the following NetView DSIPARM data set members:

- AOFOPFGW
- DSIDMNK

AOFOPFGW Modifications

In the AOFOPFGW member for each system, define the operator IDs used for both outbound and inbound gateway autotasks. If you use hardcoded passwords for your gateway autotask operator IDs, coordinate the passwords for incoming gateway autotask operator IDs, that is, operator IDs specifying other system names, with passwords specified in the ***Password*** field on the ***GATEWAY Definitions*** panel for the other systems. See *System Automation for OS/390 Defining Automation Policy* for an example.

For example, in Figure 12 on page 47, the gateway autotask definitions in AOFOPFGW on system CHI01 are:

```
GATCHI01  OPERATOR PASSWORD=RACFNNT
          PROFILEN AOFPRFAO
GATATL01  OPERATOR PASSWORD=RACFNNT
          PROFILEN AOFPRFAO
GATATL02  OPERATOR PASSWORD=RACFNNT
          PROFILEN AOFPRFAO
```

DSIDMNK Modifications

In the NetView DSIPARM member DSIDMNK, make the following modifications:

- If necessary, increase the number of cross-domain sessions by changing the CDMNSESS parameter value when defining a focal point or target system. You can do one of the following:
  - Increase the value for each definition by the number of systems that can communicate with the focal point.
  - Increase the definition values by the number of inbound operators that are defined, plus one for the outbound operator.

  The following is an example CDMNSESS statement:

  ```
  CDMNSESS  04
  ```
- Add an RRD statement (if it does not already exist) for each system with which the system being defined directly communicates. Following are example RRD statements:

  ```
  ATL01 RRD
  CHI02 RRD
  ```

For more examples of these NetView definitions, see Appendix G, “Example Automation Network Configuration”, on page 195. For more information on NetView definitions, refer to *Tivoli NetView for OS/390 Installation: Configuring Additional Components*.

Step 7D: Customize NetView for Processor Operations

Required for:

ProcOps

To enable SA OS/390, make sure that the following TOWER statements are activated in CNMSTYLE:

```
TOWER =  SA
TOWER.SA  = LICENSE
```

For SNMP and BCP Internal Interface connections, it is mandatory to make the security definitions described in "Controlling Access to the Processor Hardware Functions" on page 156.

Processor operations uses automation table entries for its operation. Make sure, the following automation table fragments are included in its master members:

ISQMSG01
Processor operations requires the automation table ISQMSG01 for its operation. This table is automatically activated when processor operations is started and deactived, once it is stopped. This automation table uses symbols defined in AOFMSGSY. Make sure this automation table contains valid definitions for the variables %AOFOPMSU% and %AOFOPNETOPER%, and that it is accessible at processor operations start time.

ISQMSGU1
This empty member is supplied by processor operations and is included in the ISQMSG01 automation table. By inserting your own automation entries or include statements of your own automation tables here, you can expand processor operations with your own automation routines which may utilize the processor operations supplied command API.

DSIDMNK
If you have defined target hardware in your processor operations configurations with an SNA based NetView connection (NVC), it is recommended to set the VTAMCP statement in your processor operations focal point NetView DSIDMNK member to `VTAMCP=NO`. This is the default when no VTAMCP statement is defined. Depending on your NetView MS environment, messages and alerts from the CPC support elements cannot be processed when the VTAMCP parameter is set to YES.

Other NetView definitions supplied by processor operations:

ISQPROF
This member contains the processor operations autotask operator profile definitions. It is available in the operator profile data set INGNPRF supplied by SA OS/390. You may customize the profile definitions in order to use OPCLASS levels different from the supplied ones. This step is optional.

Step 8: Preparing the Hardware

Required for:

SysOps and ProcOps

The steps described in this section are necessary to prepare your Hardware Management Console (HMC) and Support Elements according to the processor hardware interface you are using. For details about planning the hardware interface, refer to "Planning the Hardware Interfaces" on page 19.

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

Preparing the Hardware Management Console

Enable the HMC API and Set the Community Name

In order to control a CPC using an HMC instead of the CPC's Support Element, the Hardware Management Console API function must be enabled. If you do not plan to use the HMC to control your CPCs over the TCP/IP SNMP ProcOps interface, omit this paragraph.

1. For this task, you need to be logged on in Access Administrator mode on your HMC.
2. Select **Console Actions** and click on the *Hardware Management Console Settings* icon. On the *Settings* notebook, note the TCP/IP address of the HMC for later.
3. Select the **API** tab. If not already set, enable the API by checking the enable check box.
4. In the **Community name** field, enter a community name you have chosen. Note this community name for later.
5. Finally, select the **Apply** push button to save the changes. The message window shown informs you that the changes made require a restart of the HMC console application in order to become active.

BCP Internal Interface

To prepare the master HMC, carry out the following steps:

1. Log on to the HMC in your LAN that is to be used for change management operations with a user ID having SYSPROG authority. The HMC must have the CPC objects of your sysplex in the defined CPCs group.
2. Select **Console Actions** icon in the *Views* window and double click on the *Enable Hardware Management Console Services* icon.
3. Select the LIC change **Enabled** radio button. Select the **OK** push button to save the change, or select the **Cancel** push button if LIC change radio button was already set to Enabled.

Usually, there is one HMC in a CPC LAN environment that has LIC change permanently enabled. It will automatically be used by the BCP Internal Interface. Make sure that this HMC has all CPC objects of your sysplex in the Defined CPCs group.

SNMP

If you want to control your CPCs with the TCP/IP SNMP interface of ProcOps over an HMC, make sure its API is enabled as described in “Enable the HMC API and Set the Community Name” on page 84. Then, continue as follows:

1. Log on to the HMC in *Access Administrator* mode.
2. From the *Console Actions Work Area*, select **SNMP Configuration**.
3. Select the **Communities** tab of the SNMP Configuration notebook window.
4. For the **API** community name, enter the following information and select the **Add** push buton to add the new community name:

   Protocol
   Select UDP from the drop-down list.

   Name
   The API Community name you have chosen.

   Address
   The TCP/IP address of the Support Element which you previously made a note of.

   Network Mask
   255.255.255.255

   Access Type
   Select the **Read only** radio button.

5. For the **processor operations SNMP interface** community name, enter the information below and select the **Add** push buton to add the new community name.

   The CPC is controlled over the TCP/IP SNMP transport if it is configured for connection protocol SNMP, using the Processor (CPC) entry in the SA OS/390 Customization Dialog.

   Protocol
   Select UDP from the drop-down list.

   Name
   PROCOPS (use the community name specified in the processor entry for the CPC in your SA OS/390 policy database)

   Address
   Use the IP address of your MVS processor operations focal point system

   Network Mask
   Use 255.255.255.255 to make sure that only the addressed focal point can control the CPC. You may change the netmask to allow multiple focal point systems to control your CPC. Specify 0.0.0.0 as the address and network mask if you want to allow access from any location in your network to your CPC, using the community name from above.

   Access Type
   Select the **Read/write** radio button.

6. Select the **OK** push button to save the changed settings and close the SNMP notebook window.
7. If any of the above data was added or changed, you need to shutdown and restart the Console before the changes will be put into effect.

NVC

The following setup step is required if you want to define an SNA-based NetView connection between the Support Element of a CPC and the processor operations focal point. The CPCs that you want to connect to the processor operations focal point must have a valid SNA address configured and must be in the Defined CPCs Group of the HMC. If the CPC object is not defined to the HMC Defined CPC Group, see the HMC Object Definition task. To complete this setup:

1. Log on to the HMC with a user ID having *SYSPROG* authority.

2. From the selectable *Task List*, choose the **CPC Remote Customization** task.
3. From the *Groups View Work Area*, select the **Defined CPC Group**.
4. Mark the CPC objects you wish to select for processor operations focal point communication and click on the **Problem Management** task icon in the *CPC Remote Customization task area*.
5. In the *Problem Management* window that is displayed, select the **Enable alert generation** push button.
6. In the **Focal Point Addressing, LAN address** field, enter the 12-digit LAN address of the device serving as the gateway between the processor hardware LAN and your SNA network. This device, usually a network control unit such as a 37*xx* or 2216, must be physically connected to the processor hardware LAN.
7. Select the **Save** push button to complete the task for Problem Determination.
8. Now select **Operations Management** task and repeat these steps for the CPC objects that are still marked.

HMC Object Definition

Depending on the processor hardware interfaces, the CPCs that are to be managed must be known by the HMC, used to route OCF requests to other SEs (BCP Internal Interface), or to the HMC serving as the single point of control (SNMP), or to the HMC that routes alerts from the CPCs to the processor operations focal point (NVC).

Use the following steps to define a CPC object to an HMC:

1. Log on to the HMC with a user ID having *ACSADMIN* authority.
2. From the task list choose the **Object Definition** task. From the *Groups View* select the **Undefined CPC** group.
3. If the CPC object that you want to define to the HMC is shown in the *Undefined CPC's Work Area*, highlight it and then double click on the **Add Object Definition** task in the *Object Definition* tasks window
4. The *CPC Definition Information Notebox* is displayed, showing the available address information for this CPC object. If you do not want to change any of the address information fields or radio button settings, select the **Save** push button. For more information about the address fields or radio buttons, refer to the HMC online help
5. The CPC is now defined to the HMC. The CPC's Support Element is rebooted to activate its registration to this HMC.
6. If the CPC object that you want to define to the HMC is *not* shown in the *Undefined CPC's Work Area*, highlight the **CPC Manual Definition Template** object .
7. The *Manual Add Object Definition* window is displayed. According to your environment, choose which protocol to use for communication between the CPC's Support Element and this HMC.
8. Depending on your protocol selection, enter: An IP address; Or the SNA Network ID and CPC name; Or the token ring address of the LAN bridge in the case of an SNA connection between the HMC and the CPC over a bridged LAN.
9. Select the **OK** push button. The HMC starts to communicate with the CPC using your network information. If the Add was successful, the CPC object will be shown in the *Defined CPCs Work Area*.

Preparing the Support Elements

Before the BCP Internal Interface can be used, you need to verify for the CPC Support Elements in your sysplex that the required prerequisite MCL levels are active, and that any essential services have been enabled with the necessary settings. This requires the following:

- "Configure SNMP"
- "Enable the API and Set the Community Name" on page 88
- "Set the Cross Partition Flags" on page 88 (LPAR mode)
- "Customize the Authorization Token" on page 89

Configure SNMP

Community names have to be specified in order to use the the BCP Internal Interface transport, the TCP/IP SNMP transport for ProcOps, or both. For this task, you need to be logged on in *Access Administrator* mode on your CPC's Support Element. To complete this task:

1. Start the SNMP Configuration task by double clicking the **Console Actions** icon in the *Views* area of the Console.
2. Select the **Communities** tab of the SNMP Configuration notebook window.
3. For the **API** community name, enter the following information and select the **Add** push buton to add the new community name:

Protocol	Select UDP from the drop-down list.
Name	The API Community name you have chosen.
Address	The TCP/IP address of the Support Element which you previously made a note of.
Network Mask	255.255.255.255
Access Type	Select the **Read only** radio button.

4. If the CPC is not controlled over the BCP Internal Interface transport, omit this step.

   The CPC is controlled over the BCP Internal Interface if it is configured for connection protocol INTERNAL, using the Processor (CPC) entry of the SA OS/390 Customization Dialog.

   For the **BCP Internal Interface** community name, enter the following information and select the **Add** push buton to add the new community name:

Protocol	Select UDP from the drop-down list.
Name	SAFOS (Use the CPC authtkn name that you defined for the CPC using the customization dialogs)
Address	127.0.0.1
Network Mask	255.255.255.255
Access Type	Select the **Read/write** radio button.

5. If the CPC is not controlled over the ProcOps TCP/IP SNMP transport, omit this step.

   The CPC is controlled over the TCP/IP SNMP transport if it is configured for connection protocol SNMP, using the Processor (CPC) entry of the SA OS/390 Customization Dialog.

   For the **ProcOps SNMP interface** community name, enter the following information and select the **Add** push buton to add the new community name:

Protocol	Select UDP from the drop-down list.
Name	PROCOPS (use the community name specified in the processor entry for the CPC in your SA OS/390 policy database)
Address	*x.x.x.x* (use the IP address of your MVS ProcOps focal point system)
Network Mask	*x.x.x.x* (use **255.255.255.255** to make sure that only the addressed focal point can control the CPC. You may change the netmask to allow multiple focal point systems to control your CPC. Specify **0.0.0.0** as both the address and network mask if you want to allow access from any location in your network to your CPC, using the community name from above.)
Access Type	Select the **Read/write** radio button.

6. Select the **OK** push button to save the changed settings and close the SNMP notebook window.
7. If any of the above data was added or changed, you need to shutdown and restart the Console before the changes will be put into effect. However, before doing so, continue with the configuration steps for Console below.

For additional SNMP and API configuration information, refer to chapter ″Configuring the Data Exchange APIs″ in *zSeries 900 Application Programming Interface*.

Enable the API and Set the Community Name

In order to use the BCP Internal Interface or the SNMP interface, the Support Element API function needs to be enabled. To complete this task:

1. Start the Support Element Settings task by double clicking the **Console Actions** icon in the *Views* area of the Console.
2. Select the **API** tab of the Support Element Settings notebook window. If not already active, enable the API by checking the **Enable the Support Element Console Application Program Interface** checkbox.
3. In the **Community name** field, enter the community name you chose when you configured for SNMP.
4. Select the **Apply** push button to save the changes.
5. Finally, for the changes you have made to the Support Element to become active, you must reboot the Support Element.

Set the Cross Partition Flags

This task is only required if you use the BCP Internal Interface to connect processor hardware running in LPAR mode. For this task, you need to be logged on in *System Programmer mode* on your CPC’s Support Element. To complete this task:

1. Click on the **CPC Group** and highlight the *CPC* icon.
2. Select the **CPC Operation Customization** task.
3. Click on the **Change LPAR Security** icon. The window displayed shows the security settings from the active IOCDS for the logical partitions defined on this CPC.
4. For each logical partition that should use the BCP Internal Interface to control another partition on this CPC, check the **Cross Partition Authority** checkbox.

Customize the Authorization Token

This task is only required for NVC connections. To complete this task:

1. Log on to the Support Element with a user ID having *ACSADMIN* authority.
2. Select the **Console Actions View**. The *Console Action Work Area* is displayed.
3. Double click on the **Customize Authorization Token** icon. If you want to change the supplied default, enter a new authorization token value. The new value can be up to 8 characters long and must not contain blanks. This authorization token value must be used when defining a NVC for a CPC using the SA OS/390 customization dialogs.

Additional Verification for SEs of G3/G4 CPC Hardware

When customizing the Support Element for a G3/G4 machine, it should be verified that the SETUP.CMD file in the Support Element subdirectory C:\MPTN\BIN contains the statement `'ifconfig lo 127.0.0.1'`. Only if this statement is defined, can the BCP Internal Interface be used to target this machine.

If, after having made the neccessary HMC and SE SNMP definition steps for a G3/G4 CPC, the BCP Internal Interface communication to this hardware still fails with a condition of 0B100224, please contact your IBM customer engineer to perform the above-mentioned verification. Should the communication still fail, contact IBM software service.

Step 9: Customizing the Automation Manager

Required for:

SysOps

This step consists of the following substeps:

- "Step 9A: Customizing HSAPRM*xx*"
- "Step 9B: ARM Instrumentation of the Automation Manager" on page 90

Step 9A: Customizing HSAPRM*xx*

The HSAPRM*xx* PARMLIB member contains information required for the initialization of the automation manager and default values for other operational parameters. The member is designed to be used in common by all automation manager instances within the automation subplex.

Alternatively you can put the automation manager PARMLIB member in any partitioned data set. Then, you need to insert a statement HSAPLIB DD into the automation manager startup procedure member which refers to this partitioned data set.

A sample member called HSAPRM00 is provided in the SINGSAMP sample library. This sample is automatically copied into the PARMLIB of the automation manager (DD name HSAPLIB) when you allocate this data set as described in "Step 2: Allocate Data Sets Common to Systems" on page 67. Refer to Appendix F, "Syntax for HSAPRM00", on page 189 for the contents of this sample and the description of the parameters.

Step 9B: ARM Instrumentation of the Automation Manager

The automation manager can be enabled for Automatic Restart Manager (ARM).

A job skeleton is provided in the SINGSAMP sample library as member HSADEFA to define the SA OS/390 specific Automatic Restart Manager policy.

You can define a policy allowing you to keep the number of automation manager instances on a certain level.

In a single system environment
with more than one automation manager active, ARM can automatically restart a failing primary instance. One of the survived automation managers will take the primary role and the restarted instance will become a backup instance.

If there is only one automation manager active on a single system, ARM will automatically restart this instance again. It becomes the primary instance again and runs the takeover. The takeover time is extended by the time needed for the address space restart.

In a sysplex (subplex) environment
ARM will always restart the failing instance on the **same** system. Either there is already a backup waiting around or the restarted instance will take over.

SA OS/390 will provide a policy sample with the following major options:

- Restart only for an address space ABEND (Option ELEMTERM). The restart in case of a system breakage is not supported.

  The concept of the automation manager availability follows a ’floating’ master model. It is a peer model with one or more backup instances on different systems already active and waiting to take over. Whenever a complete system goes away the failed automation managers (backup or primary) are not restarted somewhere else.
- The ARM element name is a 16 byte string concatenation `HSAAM_sysnamexy` with:

  HSAAM_
  is a string constant as prefix

  sysname
  is the XCF member name of the automation manager which is the 8 byte MVS system name padded with ’$’, for example: MVS1$$$$

  x is a one byte digit (one of 1, 2, ... 9) automatically determined at initialization time

  y is a blank
- The restart command is the unchanged original start command, however the start mode is always HOT.
- There are no restart dependencies (no Waitpred processing)

Step 9C: Security Considerations

The job invoking the automation manager (see HSAMPROC in the sample library) must have the following access rights:

1. Access to the OMVS segment
2. It must be defined by RACF as a superuser for UNIX System Services if the automation manager will be started before JES2 initialization has completed.

3. Read access for the SYS1.PARMLIB data set
4. Write access to the log streams
5. Write access to the following data sets:
   - Trace data sets
   - Schedule override file
   - Configuration information file (DDname HSACFGIN)

Step 10: Customizing the Component Trace

Required for:

SysOps

The system operations component as well as the automation manager are using the OS/390 component trace for debugging purposes. The following setup must be done:

- Edit and - if necessary - customize the CTIHSAZZ member residing in the SINGSAMP sample library. Copy the member into SYS1.PARMLIB.
- Copy the HSACTWR member residing in the SINGSAMP sample library into SYS1.PROCLIB.
- Allocate the trace data set used by the component trace. You can use the sample job HSAJCTWR in SINGSAMP to allocate the data set. Modify the sample job where appropriate.

Note: Make sure that the job invoking the ITTTRCWR module (see HSACTWR member in the sample library) has write access to the trace output data set.

Step 11: Customizing the System Logger

Optional for:

SysOps

Note:

Though this step is optional, it is, however, recommended. The automation manager and the IBM Health Checker for z/OS and Sysplex both write history information into the OS/390 system logger and the automation agents read from it.

If you do not perform this step, users will not get any output from the INGHIST and INGHC commands.

To exploit the system logger, the following must be fulfilled:

- systems in a sysplex must run in XCF mode and

  ```
  PLEXCFG=MULTISYSTEM
  ```

  must be defined in SYS1.PARMLIB(IEASYS*xx*).

- for stand-alone systems

  ```
  PLEXCFG=MONOPLEX
  ```

  must be defined in SYS1.PARMLIB(IEASYS*xx*).

Next, the LOGR couple data sets must be formatted — if this has not already been done. For this task you can use the sample JCL provided in the HSAJFCDS member of the sample library.

To define the log stream in:

- a single system environment, use the sample JCL provided in member HSAJDLGM (for the automation manager) and INGJDLGM (for the HealthChecker)
- a sysplex, use the sample JCL provided in member HSAJDLGS (for the automation manager) and INGJDLGS (for the HealthChecker)

In both cases you may want to adapt the HLQ parameter in the LOGR policy according to your environment. The default is IXGLOGR. Use the corresponding INGJD*xxx* members as input and make the changes accordingly.

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

```
STRUCTURE   NAME(HSA_LOG)
            SIZE(8192)
            PREFLIST(cfname,cfname)

STRUCTURE   NAME(ING_HEALTHCHKLOG)
            SIZE(8192)
            PREFLIST(cfname,cfname)
```

In this CFRM policy, you have to adapt the PREFLIST for structure HSA_LOG, and ING_HEALTHCHKLOG if you are setting up the system logger for the HealthChecker. Also adapt the SIZE parameter to a recommended minimum of 8 megabytes (8M).

The system logger must be authorized. If it is not yet assigned either privileged or trusted RACF status, or both, refer to chapter *Planning for System Logger Applications* in *OS/390 MVS Setting Up a Sysplex* for more information on how to define authorization to system logger resources. The names of the system logger resources used by SA OS/390 are *HSA.MESSAGE.LOG*, *HSA.WORKITEM.HISTORY*, and *ING.HEALTH.CHECKER.HISTORY*.

The NetView agents and the automation manager address spaces need to be authorized for accessing the log streams. They need update access for

```
RESOURCE(logstream_name)
CLASS(LOGSTRM)
```

where *logstream_name* stands for *HSA.MESSAGE.LOG* as well as for *HSA.WORKITEM.HISTORY*.

In addition, all three NetView agents need update access for the log stream:

```
RESOURCE(ING.HEALTH.CHECKER.HISTORY)
CLASS(LOGSTRM)
```

For further information see section *Authorization for System Logger Application Programs* in *OS/390 MVS Setting Up a Sysplex*.

Now activate the couple data sets via the console commands:

```
SETXCF COUPLE,TYPE=LOGR,PCOUPLE=(primary_couple_data_set)
SETXCF COUPLE,TYPE=LOGR,ACOUPLE=(alternate_couple_data_set)
```

For a sysplex, after defining the new structure in the CFRM policy, activate the CFRM policy via

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=policy_name
```

Step 12: Install ISPF Dialog Panels

Required for:

SysOps, ProcOps, I/O Ops

This step is required at least on the customization focal point and on any system where you want to use the customization dialog. It is recommended that you do not move any SA OS/390 policy databases between systems because of synchronization problems.

SA OS/390 ships two types of ISPF dialogs: one for I/O operations and one for defining automation policy. The I/O operations panels are used for I/O functions. The customization dialog is used to create configuration and automation definitions.

The I/O operations and customization dialog are both invoked using the INGDLG exec. This exec provides parameters for selection of the appropriate dialogs. In addition, this exec can optionally be used to allocate the required dialog libraries. INGDLG should be invoked from an ISPF menu or from a user-defined TSO REXX exec.

Since you use the customization dialog to collect information and build control files, you normally need them only at the focal point. However, as the customization dialog allows editing of specific entry types by multiple users, you also need to observe the instructions given in the appendix *Problem Determination* in *System Automation for OS/390 User's Guide*.

The I/O operations dialogs, however, are used to input commands and get responses from the I/O operations part of SA OS/390. Since they do not support multisystem commands for I/O operations functions, you must install them on each system, focal point or target, where you wish to use them. Alternatively, you can use the workstation window set to access I/O operations function.

The INGDLG command allocates required DD names and invokes the customization dialog. Its syntax is:

Step 12: Install ISPF Dialog Panels

```
►►──INGDLG──┬─────────────────────────────────┬──┬───────────────────────────┬──────►
            │            ┌─ADMIN─────┐        │  │              ┌─YES─┐      │
            └─SELECT──(──┴─IOCONNECT─┴──)─────┘  └─ALLOCATE──(──┴─NO──┴──)───┘


            ┌─────────┐                  ┌─────────┐
            ▼         │                  ▼         │
►─DDname──(───DSname──┴──)──SYSEXEC──(───DSname──┴──)──────────────────────────────►

                                                     ┌─ING──────────────────┐
►─AOFPRINT──┬─(──DSname──)──────────────┬──HLQ──(──┴─high level qualifier─┴──)────►
            └─(──SYSOUT──(──class──)──)─┘


         ┌─none────────────────┐
►─LLQ──(─┴─low level qualifier─┴──)──┬───────────────────────────┬─────────────────►
                                     └─INITSEL──(──entrypoint──)─┘


►──┬──────────┬──────────────────────────────────────────────────────────────────►◄
   └─fastpath─┘
```

Figure 15. INGDLG Command Syntax

Its parameters are:

SELECT
Enables you to select either ADMIN or IOCONNECT. If the SELECT keyword is not specified, SELECT (ADMIN) is the default.

ADMIN
Enables the selection of automation policy dialogs. This is the default.

IOCONNECT
Enables the selection of I/O connectivity dialogs

ALLOCATE
Controls defining DD names. If ALLOCATE is not specified, ALLOCATE (YES) is the default.

YES Uses any defined DD names. This is the default.

NO Ignores any DD names that have been specified.

DDname (DSname)
Fully qualified data set name to associate. Prefixes and suffixes defined using this panel are not appended to this name. For the DD name `AOFPRINT`, the following syntax is also valid:

```
AOFPRINT(SYSOUT(class))
```

, where *class* is a valid output class, creating a DD statement with SYSOUT=class. In this case, the output is placed into the JES output class *class*.

SYSEXEC(DSname DSname DSname ...)
For DD name SYSEXEC, multiple data set names are supported: SYSEXEC(DSname DSname DSname ...). This will result in the following command:

```
TSO ALLOC ALTLIB ACTIVATE APPLICATION(EXEC)
       DATASET(DSname DSname DSname ...) UNCOND
```

AOFPRINT
For DD name AOFPRINT, *DSname* is a fully qualified data set name or a request to allocate a SYSOUT data set: AOFPRINT(SYSOUT(class))

HLQ Enables you to change the high level qualifier (HLQ) of the SMP/E data sets, which currently is ING, to a HLQ of your choice. If you do not specify this parameter, ING is retained as the default.

LLQ Enables you to establish a suffix for default data set names (The default is none).

INITSEL
may be used to provide a user-selected entry point to the customization dialog. If this keyword is specified, you will not see the ***Customization Dialog Primary Menu*** as the first panel when invoking the customization dialog, but it provides a fast path to some other panel, for example, the ***Entry Name Selection*** panel for a frequently used entry type. Valid values are those that you can specify as fast path within the customization dialog, for example, to reach the APPC application:

```
=APL; S APPC
```

or to reach application group ***CICS_APG***:

```
=APG; S CICS_APG
```

or to just reach the ***Entry Name Selection*** panel for *Application*s:

```
=APL;
```

fastpath
Any words that are not the reserved keywords. The fastpath words are passed as parameters to I/O operations dialogs, if selected.

Return codes for this routine are:

0 No errors encountered

4 ISPF is not active

8 Error in data set allocation

12 Error in data set de-allocation or a failed allocation

Step 12A: Allocate Libraries for the Dialogs

Required for:

SysOps, ProcOps, I/O Ops

To set up the dialogs, you must allocate the REXX load libraries and customization dialog load libraries. This section describes the two alternative options available:
- **Alternative 1:** Dynamic allocation of the libraries using the INGDLG exec
- **Alternative 2:** Allocation of the libraries as part of the TSO logon procedure

> **Remember:**
> Throughout "Step 12: Install ISPF Dialog Panels" on page 93 use the names of the data sets as created in "Step 3: Allocate Data Sets for the Customization Dialog" on page 68.

Alternative 1: Dynamic Allocation using INGDLG

This exec performs allocations prior to starting the dialogs. In order to invoke the exec, you need to be in ISPF. The INGDLG command parameters describe where the data sets are found. See Figure 15 on page 94 for the use of INGDLG to allocate libraries.

Note that if you use INGDLG to allocate libraries, you must still perform allocation of the ISPF product libraries as described in "Alternative 2: Add to the TSO Logon Procedure".

Alternative 2: Add to the TSO Logon Procedure

Create a new TSO logon procedure that has the SA OS/390 data sets in the appropriate concatenations.

To create a TSO logon procedure, take an existing one and modify its DD statements to include the following:

```
//ISPPLIB    DD ...
             DD DSN=ING.SINGIPNL,DISP=SHR
             DD ...

//ISPMLIB    DD ...
             DD DSN=ING.SINGIMSG,DISP=SHR
             DD ...

//ISPSLIB    DD ...
             DD DSN=ING.SINGISKL,DISP=SHR
             DD ...

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

//ISPLLIB    DD ...
             DD DSN=ING.SINGIMOD,DISP=SHR
             DD ...

//SYSPROC    DD ...
             DD DSN=ING.SINGIREX,DISP=SHR
             DD ...

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

//AOFPRINT   DD SYSOUT=...   2

//AOFIPDB    DD DSN=ING.SINGIPDB,DISP=SHR   3

//IHVCONF    DD DSN=ING.CUSTOM.IHVCONF,DISP=SHR   4
```

Notes:

1. Ensure that your ISPF temporary data sets have been allocated with enough space.
   - When a build of the automation control file is performed, each file is written to the temporary data sets before it is copied into the target data set. This

can lead to a temporary data set many thousands of lines long. For an enterprise with many processors, there may be several hundred thousand lines written to the temporary data set. These are in the ISPWRK data sets. See *z/OS ISPF Planning and Customizing* for more information, where it is recommended that you pre-allocate to VIO however, because it reduces overhead and eliminates potential problems from insufficient space.

- The ISPCTL1 temporary data set is used by SA OS/390 to hold output created by a data model report and to hold the JCL for batch submission of an ACF Build job. See *z/OS ISPF Planning and Customizing* for more information on the ISPCTL1 data set.

2. Ensure that the ISPF table output library ISPTABL is allocated. SA OS/390 uses this table output library for temporary tables created by the customization dialogs. The table output library must be a partitioned data set. The ISPTABL DD name that defines the table output library may be the same data set as the table input library, DD name ISPTLIB, but this is not required by SA OS/390. For more information, see *ISPF User's Guide*.
3. The ellipses (...) in the DD-statements indicate the presence of more information in the JCL: for example, other data sets in a concatenation.
4. User-specific data sets should be placed before the SA OS/390 data sets. Generally speaking you need to take care that the concatenation of the SA OS/390 data sets does not interfere with the concatenation with data sets from other products.
5. You should **not** include data sets for any predecessor products, AOC/MVS, TSCF, or ESCON Manager, anywhere in the concatenation.
6. The AOFTABL DD statement (**1**) is required as soon as you intend to customize your environment: this data set stores ISPF tables containing unique information created when you use the customization dialog. It is used with the customization dialog. ING.CUSTOM.AOFTABL, allocated in "Step 3: Allocate Data Sets for the Customization Dialog" on page 68, is used to hold new and modified ISPF tables created when the administrator modifies or changes the SA OS/390 Data Model or the SA OS/390 policy definitions from the SA OS/390 customization dialog. While modifications of the SA OS/390 Data Model in many customer environments need not to be changed, changes in the SA OS/390 policy definitions and ISPF tables might often occur. This makes it a required DD statement. This data set is also used to hold the data set definitions for batch processing. This data set was allocated by you in the sample INGEDLGA (see "Step 3: Allocate Data Sets for the Customization Dialog" on page 68).
7. The AOFPRINT DD statement (**2**) is used in place of SYSPRINT for IEBUPDTE, which is invoked when a user of the customization dialog creates a policy database using an SA OS/390-supplied sample as a model. If this DD statement is not allocated, SA OS/390 allocates the DD as SYSOUT=H.

   If the IEBUPDTE invocation is successful and SA OS/390 dynamically allocated the AOFPRINT file as SYSOUT=H, the output is purged. If the invocation fails, the output is saved for use in diagnosis of the problem.

   When specifying `AOFPRINT(SYSOUT(Cls))`, the output of the dynamically called IEBUPDATE utility is placed in the JES output class *Cls*. This output is not purged.
8. The AOFIPDB DD statement (**3**) points to the SA OS/390 sample library.

   The AOFIPDB DD statement is required for building system operations control files (automation control file and automation manager configuration file). It must point to a single data set, not a concatenation. In SA OS/390, this data set is required, even if you do not use any sample policy databases. AOFIPDB contains the automation manager logic deck INGLOGIC.

9. IHVCONF (**4**), is required for I/O operations. If you are not using I/O operations this DD statement is optional.
10. You should not use any DD names starting with AOF in your logon procedure except those specified in the example above. This is because the SA OS/390 customization dialog may dynamically generate AOF*xxxxx* DD names. Specifically, SA OS/390 generates AOFIN and AOFUT2 DD names.
11. I/O operations ISPF dialogs use REXX execs that invoke I/O operations commands and ISPF services. These execs must be made available to the users who want to use the ISPF dialogs. Note that the default record format of the I/O operations REXX target library (whose name is SINGIREX) is FB. The data sets in your SYSPROC concatenation might not be FB. If this is the case, the ALLOCATE command can be used, but you are not able to execute the differently formatted or sized execs. You can do one of the following to correct this:
    a. Copy the contents of the SINGIREX exec library to another data set that is already in your SYSPROC concatenation.
    b. Copy the contents of the SINGIREX exec library to a new data set that has the same characteristics as the other data sets in your SYSPROC concatenation.

If you already use a CLIST to allocate your data sets for ISPF, modify it to include the SA OS/390 data sets in the appropriate concatenations for users of the customization dialog. If you wish to create a CLIST to allocate your data sets you should find out your current allocations for the DD names that need SA OS/390 data sets allocated to them. This can be done with the LISTALC STATUS command.

Step 12B: Invoking the ISPF Dialogs

Required for:

SysOps, ProcOps, I/O Ops

The ISPF Application Selection Menu can be modified to include options for the system operations and processor operations customization dialog and for the I/O operations dialogs. These options allow a user to begin the customization dialog without having to issue commands at the TSO prompt.

Two changes are required to add the dialogs to the ISPF Application Selection Menu panel (see also Figure 16 on page 99):

adding selections to the menu

adding logic to the panel processing to invoke the appropriate dialogs.

Both sets of dialogs are invoked by the INGDLG command. Parameters of this command determine which set of dialogs is invoked.

Add the command dialogs selections to an ISPF menu panel, such as the ISPF Master Application Menu panel (ISP@MSTR) or the ISPF Primary Menu panel (ISP@PRIM).

Note: If you use a customized, non-standard ISPF primary menu panel, modify the definition for that panel instead of ISP@MSTR or ISP@PRIM.

See *z/OS ISPF Planning and Customizing* for information about customizing ISPF panels. The modified panel should be placed in a data set so that it is used by all users who have the dialog data sets in their concatenation, but it is not used by anyone who does not. You may want to copy it into an enterprise-specific panel data set that you allocate in front of your normal ISPF panel data sets. Figure 16 is an example of how a modified panel may look.

```
%-----------ISPF APPLICATION SELECTION MENU---------------------------------
%OPTION ===>_ZCMD
%                                                         +VERSION  -&ZENVIR
%  0 +ISPF PARMS - Specify terminal and user parameters   +USERID   -&ZUSER
%  1 +BROWSE     - Display source data or output listings +TIME     -&ZTIME
%  2 +EDIT       - Create or change source data           +TERMINAL -&ZTERM
&  3 +UTILITIES  - Perform utility functions              +PK KEYS -&ZKEYS
⋮
&  C +CUSTOMIZE  - SA OS/390 customization dialog
&  I +I/O-Ops    - SA OS/390 I/O Operations
&  T +TUTORIAL   - Display information about ISPF/PDF
&  X +EXIT       - Terminate ISPF using log and list defaults
%
+Enter%END+command to terminate ISPF.
```

Figure 16. ISPF Application Selection Menu

The options for the customization dialog and the I/O operations dialogs must also be added to the panel processing section of the ISPF Application Selection Menu panel as follows. The lines you add are written in italics in the example. You can select the character used to specify the dialogs on your menu.

Using TSO Logon or Your CLIST

This is the example to be followed if you allocated the data sets using the TSO logon procedure or using a CLIST of your own.

```
)PROC
&ZQ = &Z
IF (&ZCMD ^= ' ')
&ZQ = TRUNC(&ZCMD,'.')
IF (&ZQ = ' ')
·MSG = ISRU000
&ZSEL = TRANS( &ZQ
0,'PANEL(ISPOPTA)'
⋮
C,'CMD(INGDLG SELECT(ADMIN) ALLOCATE(NO))'
I,'CMD(INGDLG SELECT(IOCONNECT) ALLOCATE(NO))'
T,'PGM(ISPTUTOR) PARM(ISR00000)'
⋮
X,'EXIT'
*,'?' )
&ZTRAIL = ·TRAIL
)END
```

Using INGDLG

If you let INGDLG, described in Figure 15 on page 94, allocate the data sets dynamically prior to starting the dialogs, the following is a sample definition to be added to the ISPF processing section:

```
C,'CMD(EXEC ''ING.SINGIREX(INGDLG)'' +
  ''HLQ(MYHLQ)                      +
    AOFTABL(ING.CUSTOM.AOFTABL)      +
    SELECT(ADMIN)'')'
```

```
I,'CMD(EXEC ''ING.SINGIREX(INGDLG)'' +
  ''HLQ(MYHLQ)                       +
    IHVCONF(ING.CUSTOM.IHVCONF)      +
    SELECT(IOCONNECT)'')'
```

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

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

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

Step 12C: Reconvert I/O Operations Panels

Optional for:

I/O Ops

The I/O operations dialog panels are defined using Dialog Tag Language (DTL) for ISPF. Both the source panels and converted panels are provided in the product libraries. If you choose to update the panels, the source panels must then be reconverted.

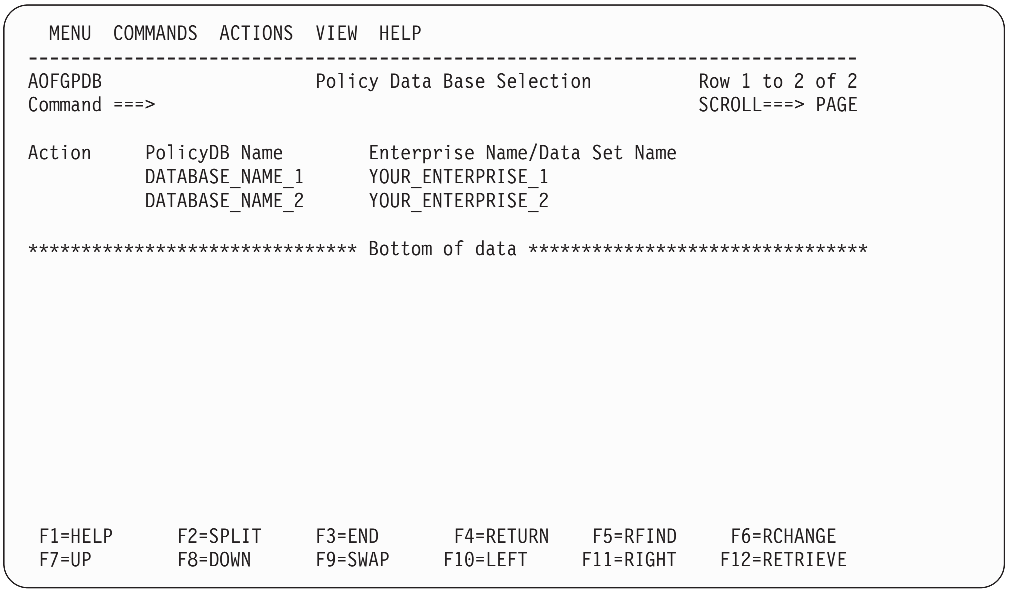
Step 12D: Verify the ISPF Dialog Installation

Required for:

SysOps, ProcOps, I/O Ops

Logon to TSO using your modified logon procedure or running your data set allocation CLIST.

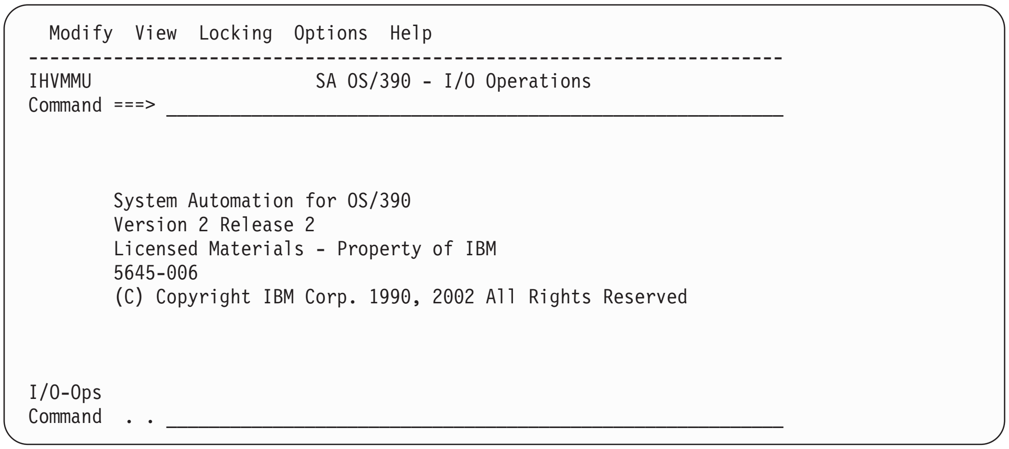
Access the customization dialog from the ISPF main menu that you defined. From the ***Customization Dialog Primary Menu*** that will appear, select option ***4 Policies*** to see a screen that looks similar to Figure 17 on page 101.



```
  MENU  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
AOFGPDB                   Policy Data Base Selection          Row 1 to 2 of 2
Command ===>                                                  SCROLL===> PAGE

Action     PolicyDB Name        Enterprise Name/Data Set Name
           DATABASE_NAME_1      YOUR_ENTERPRISE_1
           DATABASE_NAME_2      YOUR_ENTERPRISE_2

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















 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 17. Policy Data Base Selection Screen

A screen similar to the one shown in Figure 18 will be displayed if you run the REXX exec *IOCONNECT* shown on page 100. You can use the information shown to verify your SA OS/390 installation.



```
  Modify  View  Locking  Options  Help
-----------------------------------------------------------------------
IHVMMU                    SA OS/390 - I/O Operations
Command ===> __________________________________________________________



        System Automation for OS/390
        Version 2 Release 2
        Licensed Materials - Property of IBM
        5645-006
        (C) Copyright IBM Corp. 1990, 2002 All Rights Reserved



I/O-Ops
Command  . . __________________________________________________________
```

Figure 18. I/O Operations Initialization Panel

Step 13: Verify the Number of available REXX Environments

Required for:

SysOps and ProcOps

Change the value of the maximum number of available REXX environments to at least 400. The variables to do this are in the sample assembly and linkedit job in SYS1.SAMPLIB(IRXTSMPE). Change the value of the ENTRYNUM= parameter to at least 400. The sample is a user exit, so follow your SMP/E process for handling user exits. See also "Allocation Requirements for REXX Environments" on page 24.

Step 14: Customization of NetView for TEC Notification by SA OS/390

Optional for:

SysOps

This section describes the customization steps specific for TEC Notification by SA OS/390 of all involved products:

- NetView
- SA OS/390
- GEM

Depending on whether SA OS/390 messages are forwarded to a local GEM Message Adapter or a message has to be forwarded to the SA OS/390 focal point system running the GEM Message Adapter, the NetView customization is different:

- In a ***local configuration***, there is only one operator and you can use the default operator ID AUTOTEC.
- In a ***distributed configuration***, you need to define a different operator ID on each target system. If the focal point is also configured as a target system that triggers messages and alerts, you need to define another different operator ID on the focal point itself. In case of a distributed configuration, you need to adapt the synonym table.

  All operator IDs of all target systems must be defined on the focal point.

Review the synonyms for TEC Notification by SA OS/390 and set all listed synonyms to their appropriate value.

- %AOFTECTASK% and %AOFTECTASKQ%

  This is the name of the autotask for sending SA OS/390 events to the Tivoli Enterprise Console. It is the operator ID you defined in your configuration. The default is `AUTOTEC`.
- %AOFTECPPI%

  This is the NetView PPI Receiver ID of the GEM message adapter (with quotes). The default is `IHSATEC`.
- %AOFTECMODE%

  This is the event generation mode (with quotes). Possible values are:
  - LOCAL: the GEM message adapter is running on **this** system. LOCAL is valid for the ***local configuration*** (“Environment Configurations” on page 37) and for the focal point in the ***distributed configuration***.
  - REMOTE: the GEM message adapter is running on a remote automation focal point. SA OS/390 messages will be generated on **this** target system and forwarded to a **remote** automation focal point system. There is no local GEM message adapter which can process SA OS/390 messages. REMOTE is valid for the target system in a ***distributed configuration*** (“Environment Configurations” on page 37).

  The default is `LOCAL`.

Modifying Existing Files

Table 16 shows all product files which need to be modified.

Table 16. Product Files to be Modified

File Name	DD Name	Description
AOFMSGSY	DSIPARM	synonyms used within the automation table
AOFOPF	DSIPARM	operator definitions

AOFMSGSY

Locate the automation fragment `AOFMSGSY` to update the required synonyms.

Customizing the Auto Operators Policy Object

Define the auto operator AUTOTEC using the SA OS/390 customization dialog **Automation Operator Definitions**.

For a complete description of the required dialogs, refer to *System Automation for OS/390 Defining Automation Policy*.

Customizing the System Policy Object

You must define INGMTEC as additional message table, using the SA OS/390 customization dialog **Environment Setup** (policy selection AUTOMATION SETUP).

For a complete description of the required dialogs, refer to *System Automation for OS/390 Defining Automation Policy*.

Removing Messages

You may want to remove a message from the set of mapped messages. To do this, you only need to remove the *IF ... THEN* statement corresponding to the affected message from file *INGMTEC*.

Customization of GEM

This section describes how to customize GEM, namely the message adapter and the alert adapter for messages and alerts from SA OS/390.

Modifying GEM Files

Table 17 shows all product files which need to be modified.

Table 17. Product Files to be modified

Member Name	DD Name	Description
IHSAINIT	IHSSMP3	initialization file for the GEM Event Service
IHSAMCFG	IHSSMP3	message adapter configuration file
IHSAACFG	IHSSMP3	alert adapter configuration file
IHSAMFMT	IHSSMP3	message adapter format file
IHSAACDS	IHSSMP3	alert adapter CDS file
IHSAECFG	IHSSMP3	Event Receiver Configuration file

Step 14: Customization of NetView for TEC Notification by SA OS/390

The following is a brief list of steps needed to customize the GEM Event/Automation Service for SA OS/390 specific message/alert routing. For detailed guidance see ***Tivoli GEM User's Guide chapter Event/Automation Service***.

1. Adapt the initialization file for the GEM Event/Automation Service IHSAINIT. At least the following values have to be defined:
   - *ALRTCFG*: specifies the alert adapter configuration, for example: `ALRTCFG=IHSAACFG`
   - *MSGCFG*: specifies the message adapter configuration file, for example: `MSGCFG=IHSAMCFG`
   - *PPI*: specifies the PPI receiver ID used by the GEM Event/Automation Service, for example: `PPI=IHSATEC`

   The values given here are examples only. They will be further used within this chapter.

   Note: The *PPI* receiver ID for the GEM Message Adapter (PPI) specified here, must be the same as the one defined in the synonym section of the NetView message automation table.
2. Adapt the message adapter configuration file IHSAMCFG. You must at least define: ServerLocation=*IP address of your Tivoli Enterprise Console*
3. Adapt the alert adapter configuration file IHSAACFG. You must at least define: ServerLocation=*IP address of your Tivoli Enterprise Console*
4. Insert the include statement shown in Figure 19 at the end of the GEM Event/Automation Service format file IHSAMFMT. This will activate the message/event mapping defined in the message adapter format file *INGMFMT* for SA OS/390 messages.

```
/* ------------------------------------------------------------------ */
/* System Automation for OS/390 (AOF) message to T/EC event mapping   */
/* ------------------------------------------------------------------ */
%INCLUDE INGMFMT
```

Figure 19. Format File Include Statement

5. Insert the include statement shown in Figure 20 into the GEM Event/Automation Service CDS file IHSAACDS to activate the alert / event mapping. Make sure the SA OS/390 specific statements precede the more general statements of the same class. This can be achieved by inserting the include statement at the top of file *INGACDS*.

```
/* ------------------------------------------------------------------ */
/* System Automation for OS/390 (AOF) alert to T/EC event mapping     */
/* ------------------------------------------------------------------ */
%INCLUDE INGACDS
```

Figure 20. CDS File Include Statement

6. Adapt the Event Receiver Configuration file IHSAECFG. You must at least define: *NetViewAlertReceiver=NETVALRT*

Step 15: Compile SA OS/390 REXX Procedures

Optional for:

SysOps and ProcOps

You should perform this step to gain considerable performance improvement for system operations startup.

You can optionally compile the SA OS/390 automation procedures, which are written in REXX. The decision to compile the SA OS/390 automation procedures implies an added responsibility for recompiling whenever ING.SINGNREX members are affected by SMP/E maintenance. To compile and execute these automation procedures, the IBM Compiler and Library for REXX/370 must be installed on your system along with their prerequisite products.

The JCL job INGEREXR and related routine INGEREXC are provided in the SA OS/390 sample library to help you compile the ING.SINGNREX members. Modify the data set names and jobcard in INGRXRUN as necessary and submit the job. Specify the name of the resulting compiled-REXX data set in place of ING.SINGNREX in your NetView application startup procedure.

Consult the *REXX/370 User's Guide and Reference R3* (SH19-8160) for the compiler options that apply to your installation. If necessary, change the INGEREXC routine accordingly.

Step 16: Defining Automation Policy

Required for:

SysOps and ProcOps

Before you can start using automation, you need to define your automation policy using the customization dialog. This involves the following actions:

- If applicable, migrate/merge existing policy information; you can use the sample job INGEBMIG in the SINGSAMP sample library.
- Add further policy definitions
- Build a new policy database
- Distribute the policy definitions (the policy database) where required.

If you start from scratch, use the IBM samples delivered with the product and create your new policy database. In such a case, read the information in the section *Creating a New Policy Database* in *System Automation for OS/390 Defining Automation Policy*.

Then invoke the customization dialog to define your automation policy. Start by defining the following policy objects:

- Applications
- Application groups

- Processors
- Systems
- System defaults
- A group for each sysplex

You will find detailed information on how to perform these steps in *System Automation for OS/390 Defining Automation Policy* which provides information on using the customization dialog for the required definitions.

Step 16A: Build the Control Files

When the policies for the SA OS/390 components have been defined, use the BUILD command to create the system operations control files (automation control file and automation manager configuration file, needed for automation), processor operations control file and NetView operator definitions. The BUILD command is available from various panels of the customization dialog. For more information on how to perform this step, refer to the manual *System Automation for OS/390 Defining Automation Policy*. You can use the sample job INGEBBLD in the SINGSAMP sample library.

Note:

It is mandatory to use the SA OS/390 customization dialog to create policy objects for the resources you want to automate. Do not edit the automation control files (ACF) manually.

A manually edited automation control file cannot be used to start SA OS/390.

Step 16B: Distribute System Operations Control Files

The system operations control files consist of the automation control file and the automation manager configuration file. You need to make the control files available to the automation agents and automation managers on the target systems. All automation managers and automation agents within the same sysplex must have access to the same system operations control files or a copy of them. You must send the files to the target sysplexes and make the data available to the automation agents and the automation managers. For the automation agents, it can either be in the DSIPARM concatenation or in a separate data set which has the same name as that known to the automation manager. For the automation managers it can either be placed within the automation managers' current configuration data set or the automation managers can be told to use a new configuration data set.

Step 17: Define Host-to-Host Communications

Required for:

SysOps, ProcOps, I/O Ops

VTAM definitions are required for both host-to-host communications and host-to-workstation communications. This section of the installation addresses the host-to-host communications.

Verify that your NetView APPL member is consistent with the steps that follow.

The host-to-host communications require:
- Defining each host as a CDRM
- Defining the host ACB

Step 17A: Customize the SYS1.VTAMLST Data Set

Required for:

SysOps and ProcOps

Edit the member that defines NetView to VTAM and do the following:

1. Include as many NetView operator subtask APPL statements as you defined operators in the DSIOPF member of the NetView DSIPARM data set.
2. Include as many terminal access facility (TAF) APPL statements as there are NetView operator subtask APPL statements. There are two types of TAF APPL statements: operator and full-screen. The number of TAF APPL statements for each type must match the number of NetView operator subtask APPL statements. See *Tivoli NetView for OS/390 Installation: Configuring Additional Components* for more details.
3. Define the NetView primary program operator interface task (PPT) as AUTH=(NVPACE,SPO). This causes unsolicited VTAM messages to be broadcast on the SSI and therefore be available to NetView. If, however, you have another NetView defined as a primary program operator application program (PPO), then it receives unsolicited messages first and messages do not reach the secondary program operator application program (SPO) defined NetView. See *Tivoli NetView for OS/390 Installation and Administration* for information on PPO and SPO definitions.

For each target hardware, defined with an SNA based NVC connection to the processor operations focal point, VTAM majornode definitions are required to enable the hardware access for processor operations. Appendix D, “Processor Operations Sample”, on page 181 illustrates this for an OSA adapter being the SNA gateway for the Support Elements and the definition of an SE as a VTAM Switched Majnode. For other VTAM definition examples, refer to *Managing Your Processors*, (GC38-0452-08).

Step 17B: Perform VTAM Definitions

Required for:

I/O Ops

Note: This applies to I/O operations host-to-host communications only. If you have configured a prior level of ESCON Manager, these definitions remain the same.

Step 17: Define Host-to-Host Communications

In order to use VTAM for I/O operations, there are some definitions that VTAM requires. These definitions are in addition to those needed for the installation and running of VTAM. If you already have VTAM installed, some of these definitions may already exist.

The I/O operations program in each host that carries on this communication must be defined as a VTAM application in each host. The I/O operations program that it communicates with in another host must be defined as a cross domain resource. I/O operations uses the LU 0 protocol for the communication between hosts and the LU 6.2 protocol for host-to-workstation communications.

Since the means of the I/O operations program may be a channel-to-channel adapter, this connection has to be defined to VTAM via VTAM definition statements.

If the alternate path used is via a network communications program (NCP), then the NCP must be defined to VTAM.

In order for VTAM, to choose what routes to use for this communication and what priorities to assign, PATH statements and CLASS OF SERVICE must be defined.

An example of some of these VTAM definition statements will be shown in the following picture as an example.

```
+----------+                +----------+
|          |                |          |
|   V10M   |                |   V20M   |
|          |                |          |
+----------+                +----------+
| IHVAPPL1 |<-------> CTC <----->| IHVAPPL2 |
+----------+                +----------+
```

Figure 21. VTAM Definition Statements

In this example, there are two hosts running I/O operations. One application is named IHVAPPL1 and is in subarea 10. The second application is named IHVAPPL2 and is in subarea 20. Each host has its own set of VTAM definition statements.

```
V10M                             V20M
____                             ____

      VBUILD  TYPE=CDRM    |
V10M  CDRM    SUBAREA=10   |---> Same statements in this host.
V20M  CDRM    SUBAREA=
```

The appropriate definitions are needed for each host that will be communicating via I/O operations. Each host will be defined as a CDRM.

If a communication path between the hosts is a channel to channel adapter, this has to be defined to VTAM.

Note: Change each “*x*” to the appropriate value.

```
CTCV20  VBUILD  TYPE=CA
label1  GROUP   LNCTL=CTCA,
                DELAY=x,
                MIH=x,       (cause link to inop if
                              sio timeout occurs)
                REPLYTO=x    (tells vtam how long to
                                wait for completion after
                                channel program started)
      label2  LINE    ADDRESS=x,    (channel unit address of
                                channel to channel adapter)
                      MAXBFRU=x     (# of buffers vtam will use to
                                receive data)
      label3  PU      PUTYPE=4,TG=1
```

Each I/O operations program must be defined via an application statement in each host. The customer specified names must be unique in the network. These are the names by which each I/O operations will be known by the other I/O operations hosts.

The ACBNAME parameter is required for I/O operations this name must be IHVISC, and this name must be reserved for this use only.

The parameters SONSCIP=YES and AUTH=ACQ must also be specified.

For I/O operations it is strongly recommended that the DLOGMODE parameter and the MODETAB parameter given in the example, or equivalent definitions should be used. For your information: A RUSIZE of ′zero′ is used with this LU TYPE 0 protocol.

```
       VBUILD TYPE=APPL         |        VBUILD TYPE=APPL
IHVAPPL1 APPL ACBNAME=IHVISC,   | IHVAPPL2 APPL ACBNAME=IHVISC,
              AUTH=ACQ,         |               AUTH=ACQ,
              SONSCIP=YES,      |               SONSCIP=YES,
              DLOGMOD=INTERACT, |               DLOGMOD=INTERACT,
              .                 |               .
              .                 |               .
              MODETAB=ISTINCLM  |               MODETAB=ISTINCLM
```

Using the above VTAM definitions the LOGMODE table entry would be,

```
IBM3767  MODEENT LOGMODE=INTERACT,FMPROF=X'03',TSPROF=X'03',
PRIPROT=X'B1',SECPROT=X'A0',COMPROT=X'3040'
```

Each host must have a cross-domain definition for the other I/O operations host applications. They are defined as cross domain resources.

```
VBUILD  TYPE=CDRSC              |         VBUILD  TYPE=CDRSC
IHVAPPL2 CDRSC  CDRM=V20M       | IHVAPPL1 CDRSC  CDRM
M
```

The communication paths between the I/O operations hosts must be defined.

```
PATH  DESTSA=20       |          PATH  DESTSA=10
ER0=(20,1)            |          ER0=(10,1)
ER1=(20,1)            |          ER1=(10,1)
VR0=1                 |          VR0=1
VR1=0                 |          VR1=0
```

The CLASS OF SERVICE definition:

```
ISTSDCOS COSTAB               | ISTSDCOS COSTAB
         .                    |          .
         .                    |          .
         .                    |          .
IHVAPPL1 COS VR=((0,2),(1,2)) | IHVAPPL2 COS VR=((0,2),(1,2))
         .                    |          .
         .                    |          .
         COSEND               |          COSEND
```

In addition to the VTAM definitions, you need to define the APPC/MVS environment to allow I/O operations functions on the focal point to communicate with I/O operations functions at the SA OS/390 workstation. Refer to *Multiplatform APPC Configuration Guide* for help in doing this.

Step 18: Add Additional VTAM APPL Statements

Required for:

SysOps and ProcOps

In your VTAM definitions, verify that there are enough VTAM APPL statements to allow all of the operators to be logged on. You should have one statement for each person who is logged on as a NetView operator. Do not include automation operators. If you add VTAM APPL statements, be sure to increase the POSPOOL value in DSIDMN appropriately.

Step 19: Enabling SA OS/390 to Restart Automatic Restart Manager Enabled Subsystems

Optional for:

SysOps

If you intend to use the OS/390 Automatic Restart Manager (available with MVS 5.2 and above) and you wish to coordinate its actions with those of SA OS/390, you must ensure the following:

- The SA OS/390-supplied element restart exit (ERE) must be available to OS/390. The exit, AOFPERRE, is in the ING.SINGMOD2 data set. No customization is required.
- The AOFARCAT autotask must be created. The autotask name is included in the AOFOPF member and is created automatically by NetView if you install SA OS/390 without changing AOFOPF.
- The NetView Subsystem Support Interface (SSI) must be active for the coordination of SA OS/390 and OS/390 automatic restart management to occur.
- As part of its Automatic Restart Manager support, SA OS/390 claims all PPI receiver IDs starting with AOF. If you have any other PPI receivers named AOF*xxxx*, results are unpredictable.

For further information on the relationship between key; and Automatic Restart Manager, see *System Automation for OS/390 Defining Automation Policy*.

Step 20: Define Security

Optional for:

SysOps

You should perform this step if you want to ensure that only authorized staff can manage the resources in your environment.

When you define your SA OS/390 environment, you need to ensure that both your operations staff and your automation facilities are authorized to manage the resources in that environment. You can control human and automation operator authority using the NetView operator definition file (DSIOPF) or a security product, such as RACF. Both of these facilities provide password security to control access by operators.

The following is recommended:

- Use NetView facilities to limit the use of commands and keywords to authorized operators or to limit (to specific systems) an operator’s span of control
- Control access to the NMC workstation by a user ID, password, and RODM access information
- Include logic in the NetView automation table or in your automation routines to verify the source of a message before taking an action
- Use RACF to limit the use of OS/390 system commands to authorized operators

Refer to “Operator Profiles” on page 145 for details.

For SNMP and BCP Internal Interface connections, it is mandatory to make the security definitions described in “Controlling Access to the Processor Hardware Functions” on page 156.

For UNIX System Services automation, one or more UNIX segments (OMVS) must be defined. For details, refer to “Defining the UNIX Segments (OMVS) for UNIX System Services Automation” on page 158.

See “Installing the SA OS/390 Password Protection Feature (Optional)” on page 160 for more information about installing the SA OS/390 Password Protection Feature, which is optional. I/O operations command security definitions are found in “Step 6 — Define a RACF Profile for I/O Operations” on page 163.

Step 21: Customize the Status Display Facility (SDF)

Optional for:

SysOps

If you decide to use SDF as the SA OS/390 full-screen operator interface for monitoring automated resource statuses at the NetView 3270 console, then customizing SDF involves defining the following:

- SDF initialization parameters. These are defined in the AOFINIT member of a NetView DSIPARM data set.
- Resource hierarchy or tree structure. The AOFTREE member of a NetView DSIPARM data set includes the appropriate tree members, which contain the resource hierarchy information.
- Color and priority assignments for resource status types. These have default values set up by SA OS/390 (see *System Automation for OS/390 User’s Guide* for details), but overrides to color and priority assignments can be defined by the user using the SA OS/390 customization dialog.

- SDFROOT. A root name for the SDF tree can be specified on the Environment Setup Panel of the customization dialog. If you do not specify a new root name, it defaults to the value specified for SYSNAME.

See *System Automation for OS/390 Customizing and Programming* for detailed information about customizing SDF.

Step 22: Check for Required IPL

Required for:

SysOps, ProcOps, I/O Ops

An IPL is only required if:

- in "Step 4D: Update LPALST*xx*" on page 72 you did ***not*** decide to use the solution to dynamically add the modules to the LPALST
- in "Step 4E: Update LNKLST*xx*" on page 72 you updated LNKLST and you did ***not*** decide to use the solution to dynamically add the modules to the LNKLST
- "Step 4F: Update IEFSSN*xx*" on page 73 has been required because the IEFSSN*xx* member has not been updated during NetView installation and you cannot use the OS/390 command SETSSI for a dynamic update of the subsystem name table.

Step 23: Automate System Operations Startup

Required for:

SysOps

Add commands to the COMMND*xx* member of SYS1.PARMLIB to start the automation NetView when OS/390 starts. You may also need to modify an IEASYS*xx* member of SYS1.PARMLIB to specify which COMMND*xx* or other PARMLIB members to use during IPL. SA OS/390 initialization begins with starting system operations. If an SA OS/390 automation policy is used, system operations subsequently starts processor operations and I/O operations.

Make the described changes to the following SYS1.PARMLIB data set members:

COMMND*xx*

Make sure that the procedure names you choose match those specified in the SYS1.PROCLIB data set.

Compare the contents of the COMMND*xx* member with the INGECOM member which resides in the SINGSAMP sample library. Edit the COMMND*xx* member and do the following:

1. If you want to use the recording of IPL function (INGPLEX IPL command) add the following statement in the COMMND*xx* member:

```
COM='S HSAPIPLC,SUB=MSTR'
```

This procedure collects the IPL information in MVS.

2. Ensure that an appropriate MPFLST member is specified. The AOFCOM member contains a SET MPF statement that specifies the suffix SO. This identifies MPFLSTSO as the MPF list member OS/390 is to use. (You can use any two characters *xx* as a pointer to your MPFLST*xx* member. It is helpful if you use the same suffix you used to name other PARMLIB members. For example, see the statement `SET PROG=xx` described in "Step 4A: Update PROG*xx*" on page 70).
3. If you are running more than one NetView on your system, ensure that you have included start commands for the Automation NetView.

```
COM='S NETVSSI,SUB=MSTR'
COM='S NETVSTRT,SUB=MSTR'
```

Note:

NETVSSI here is a placeholder for the name of the member to which you copied the NetView subsystem interface startup procedure in "Step 6: Customize SYS1.PROCLIB Members" on page 77.

NETVSTRT here is a placeholder for the name of the member to which you copied the NetView application startup procedure in "Step 6: Customize SYS1.PROCLIB Members" on page 77.

This adds commands that select the correct MPF entries and that start NetView.

IEASYS*xx*

Edit the IEASYS*xx* member to specify which SYS1.PARMLIB data set members to use during the IPL process. This is done by specifying the 2-character suffix of the SYS1.PARMLIB member names. If you choose SO, then the statements in the IEASYS*xx* member would be as follows:

```
APF=SO
CMD=SO
CON=SO
SSN=SO
SCH=SO
LNK=SO
LPA=SO
```

For example, because APF=SO, the system uses the IEAAPFSO member during the IPL process.

How to Automate the Automation Manager Startup

Note: The system on which the automation manager should be started must be defined as policy object *System* in the policy database which will be used to create the automation manager configuration file that this automation manager uses (see also "Step 16A: Build the Control Files" on page 106.

To enable automatic startup of the automation manager whenever SA OS/390 is started, add the start command for the automation manager

```
S procname,SUB=MSTR
```

to the COMMND*xx* PARMLIB member, where *procname* is your selected name of the automation manager start procedure.

Step 23: Automate System Operations Startup

You can find a sample startup procedure called HSAMPROC in the SINGSAMP sample library, so that your entry in the COMMND*xx* member could look as follows:

Sample COMMNDxx entry

```
'S HSAMPROC,JOBNAME=HSAM&SYSCLONE.,SUB=MSTR'
```

How to Automate MQSeries Startup

When you use MQSeries for manager-agent communication and status backup, you can automate MQSeries and let it be started and stopped by SA OS/390 (for details on how this is made possible, see “Exploiting MQSeries V5R2” on page 30), and this is the recommended way.

In a full sysplex environment it is recommended that you start the local MQSeries manager with its associated DB2 simultaneously immediately after JES is up and running. In a single system case with MQSeries version 2.1, DB2 is not needed. See the related product installation manuals for information on how to start MQSeries and DB2 and define these resources to SA OS/390 in the customization dialog. For more information, also refer to “Peer Recovery Considerations” on page 33.

Consider that the subsystem RRS is also necessary for shared DB2 database functions.

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

Step 24: Verify Automatic System Operations Startup

Optional for:

SysOps

After you have installed the host components of SA OS/390, it is recommended that you perform the following steps for verification purposes:

1. Perform an IPL, if you have not done this according to “Step 22: Check for Required IPL” on page 112. Then start SA OS/390 and perform a coldstart. A coldstart is performed by default unless you specify the warmstart option. Do not select the warmstart option, because you might have policy data from earlier releases in your warmstart cache.

   The following messages should appear on the system console:

   ```
   AOF532I hh:mm:ss AUTOMATION ENVIRONMENT HAS BEEN INITIALIZED
   AOF540I hh:mm:ss INITIALIZATION RELATED PROCESSING HAS BEEN COMPLETED
   ```

2. Use the NetView LIST command to confirm that the following SA OS/390 tasks are active:

Task Name	Description
AOFTSTS	automation status file task
INGPXDST	XCF communication task

To confirm that these tasks are active, log on to NetView, and enter the NetView LIST command to display the status for each task:

```
LIST taskname
```

3. Use the commands INGAMS and INGLIST to verify that they work.
4. Use the SA OS/390 DISPSTAT command in NetView to confirm that subsystem status and automation flag settings are what you expect. Enter the DISPSTAT ALL command to display the status of automated subsystems and automation flag settings: See *System Automation for OS/390 Operator's Commands* for information about the DISPSTAT command.
5. Use the SA OS/390 DISPAUTO command in NetView to display a menu that allows you to initiate further command dialogs. These display information about your automation. Enter DISPAUTO and then choose one of the menu options. See *System Automation for OS/390 Operator's Commands* for information about the DISPAUTO command.
6. Confirm that the automation shuts down and restarts the subsystems as you expect. You can shutdown and restart each automated resource individually using the following SA OS/390 command:

   ```
   INGREQ resource REQ=STOP SCOPE=ONLY RESTART=YES
   ```

   If any of the resources (subsystems) do not restart as you expect, make corrections to your automation policy.

Step 25: Install an SA OS/390 Satellite

Optional for:

SysOps

This step is only required if your enterprise runs one Automation NetView and one Networking NetView with GMFHS on the focal point system or on another focal point NetView. Then you must install SA OS/390 on the automation NetView used for system automation.

Step 25A: Customize the Networking NetView or Focal Point NetView Startup Procedure

In SYS1.PROCLIB or another procedure library, find members used to start the Networking NetView application. Insert the data set names from the following table into the indicated DD concatenations.

Notes:

1. The data sets listed in Table 18 should appear last in your concatenation. If they appear before other data sets (for example, data sets containing members customized for automated network operations [AON/MVS]), results are unpredictable.
2. The *ING.SINGMOD1* library needs to be authorized for *APF*.

Table 18. Members to start the Networking NetView

DDNAME	System Operations Data Set
STEPLIB	ING.SINGMOD1
DSICLD	ING.SINGNREX

Table 18. Members to start the Networking NetView (continued)

DDNAME	System Operations Data Set
DSIPARM	ING.SINGNPRM
DSIMSG	ING.SINGNMSG
DSIPRF	ING.SINGNPRF
CNMPNL1	ING.SINGNPNL

Step 25B: Customize the Networking NetView or Focal Point NetView DSIPARM Data Set

Several members in the DSIPARM concatenation must be customized for the SA OS/390 satellite. Before editing an SA OS/390 member, remember to copy it from ING.SINGNPRM into a new, user-defined data set that is placed before ING.SINGNPRM in the concatenation.

CNMSTYLE

To enable SA OS/390, make sure that the following TOWER statements are activated in CNMSTYLE:

```
TOWER =  SA
TOWER.SA  = SATELLITE
```

AOFMSGST

If you do not choose to use the NetView operator IDs defined by SA OS/390, copy and edit AOFMSGST to contain the appropriate definitions of the synonyms %AOFOPMSU%, %AOFOPHB% for your Networking NetView. %AOFOPMSU% is a synonym for the operators that can be routed commands as a result of alerts trapped in the NetView automation table. %AOFOPHB% is a synonym for the operator that can be routed heartbeat alerts trapped in the NetView automation table. (Note that there can be only one operator defined for %AOFOPHB% and it must be unique and not used for any other functions). Other synonyms in the member are not specific to the Networking NetView environment.

AOFRODM

Copy and edit AOFRODM to contain the correct name for your RODM and a user ID authorized to update it.

Step 26: Installing and Customizing the NMC Focal Point

Required for:

SysOps

The SA OS/390 topology manager extracts resource topology information from one or more automation managers and maintains corresponding objects within RODM. This information is available if you have completed the previous installation steps. The following sections describe how to customize the SA OS/390 topology manager for the operators.

Step 26A: Preparing for NMC

Optional for:

SysOps

Make sure you have RODM, GMFHS, and MultiSystem Manager installed and working. For information on how to do this, refer to *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide*, *Tivoli NetView for OS/390 Graphic Monitor Facility User's Guide* and *Tivoli NetView for OS/390 MultiSystem Manager User's Guide*.

The start and dependency sequence of the related components can be defined using the customization dialog as follows:

1. Define RODM as application.
2. Define the RODM Loader as TRANSIENT using a HASPARENT relationship to RODM to assure the start only if RODM is completely up and running.
3. Define GMFHS as application using a HASPARENT relationship to the RODM Loader to assure the start only if the RODM Loader has ended correctly.

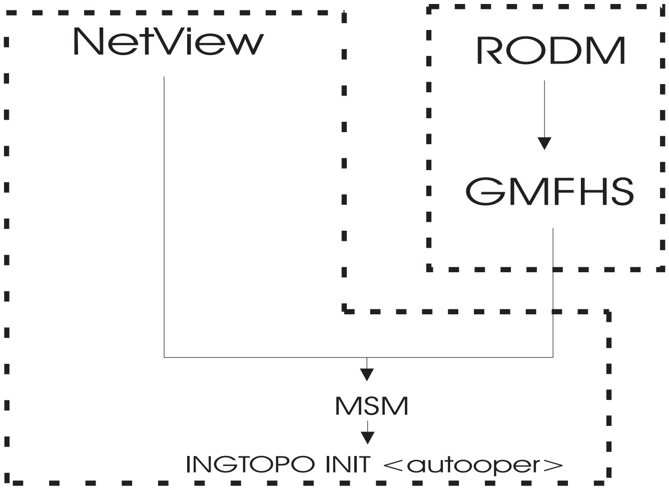


Figure 22. Environment for the SA OS/390 Topology Manager

Figure 22 shows the dependencies needed to run the SA OS/390 topology manager: NetView and MultiSystem Manager on one hand run in a common address space, RODM and GMFHS on the other hand run in a separate address space. MultiSystem Manager depends on NetView and on GMFHS. The SA OS/390 topology manager depends on MultiSystem Manager.

SA OS/390 ships a NetView message automation table fragment AOFMSGST that automates this setup. This fragment must be defined in the customization dialog to be loaded on the focal point only. With this table, the SA OS/390 topology manager is started after the completion message from MultiSystem Manager.

For additional information, see also the description of CNMSTYLE in “Step 25B: Customize the Networking NetView or Focal Point NetView DSIPARM Data Set” on page 116.

For security considerations, refer to “Securing Focal Point Systems and Target Systems” on page 145.

The RODM name must be customized in member AOFRODM on the focal point system (see “Step 25B: Customize the Networking NetView or Focal Point NetView DSIPARM Data Set” on page 116), customizing the RODM name on any other system is not necessary.

Note: To activate the Alert Forwarding process on the target system, set the AAO AOFSENDGMFHSREQUEST in AOFEXDEF accordingly. (See Appendix B: Global Variables to Enable Advanced Automation in *System Automation for OS/390 Customizing and Programming*.)

Step 26B: Modify the NetView DSIPARM Data Set for the SA OS/390 Topology Manager

There are a few things you have to do to prepare for the SA OS/390 topology manager to run. Table 19 lists the data sets to be modified for this purpose.

Table 19. DSIPARM Members to be modified for the SA OS/390 Topology Manager

DSIPARM Member	Description
DSIDMNK	NetView system level parameters for NetView initialization
DSI6INIT	Initialization member for the NetView DSI6DST task.
AOFOPFFP	system operations automation operator definitions
DSICRTTD	NetView CNM router initialization member
DUIFPMEM	NetView focal point definitions
DUIGINIT	GMFHS initialization member
FLCSAINP	MultiSystem Manager initialization member
INGTOPOF	NMC definition member

DSIPARM.DSIDMNK

To avoid further changes, alert forwarding `ALERTFWD NV-UNIQ` is recommended. However, any of the following SNA-MDS settings can be defined:

- ALERTFWD SNA-MDS=LOGONLY
- ALERTFWD SNA-MDS=AUTHRCV
- ALERTFWD SNA-MDS=SUPPRESS

While SNA-MDS is not absolutely required, it might be important as it allows the construction of networks with intermediate focal points and hot backups.

If the network contains an intermediate focal point, then ALERTFWD SNA-MDS must be specified in DSIDMNK. If the network does not contain an intermediate focal point, then ALERTFWD NV-UNIQ may be specified in DSIDMNK.

If ALERTFWD SNA-MDS is specified in DSIDMNK, the following entries must be added to sample BNJRESTY:

```
E0 AUTO  SYSTEM AUTOMATION FOR OS/390
E1 DOMN  SYSTEM AUTOMATION FOR OS/390
E2 NET   SYSTEM AUTOMATION FOR OS/390
```

Note: The three values shown above (’E0’,’E1’, and ’E2’) are the first three user-defined values. If you already have user-defined entries in BNJRESTY, you may use alternative values for these entries.

For more information on how to add user-defined entries (E0 - EF) to BNJRESTY, refer to the following chapters in *Tivoli NetView for OS/390 Customization Guide*:

- Customizing Hardware Monitor Displayed Data
- Using NMVT Support for User-Written Programming
- Adding or Modifying Resource Types

For more information about the ALERTFWD statement, refer to *Tivoli NetView for OS/390 Administration Reference*.

DSIPARM.DSI6INIT

This is the initialization member for the NetView DSI6DST task and needs to have the appropriate focal point defined.

```
     DEFFOCPT TYPE=ALERT,PRIMARY=NETA.CNM02,BACKUP=NETA.CNM03
```

Note that on the focal point and the backup you will need different members, as NetView complains if a definition references its own system.

Usage of the LU 6.2 alert forwarding mechanism allows for the construction of focal point networks that include intermediate focal points.

Autotask Operator IDs

Each focal point that will be running the SA OS/390 topology manager must have an autotask defined for it. Your environment may have one or more of the following types of focal point:

- the primary focal point
- the secondary focal point
- the intermediate focal point (IFP)

This requires a definition in DSIPARM.DSIOPF:

```
  &domain.TPO   OPERATOR    PASSWORD=&domain.TPO
               PROFILEN    AOFPRFAO
```

This definition must be made on the focal point(s) and on each target system. It should only be started as an autotask on the focal point.

An include member, DSIPARM.AOFOPFFP, has been provided to help you centralize and manage these operator IDs. You need to customize it to contain the operator IDs for your focal points.

The &domain. variable contains the focal point’s domain ID. This is just a suggestion for the naming scheme.

Note: The names must be unique on the focal point and the target systems.

Additionally, on the focal point, the operator ID must be defined in the DSIPARM.AOFMSGST member, as the value for the %AOFOPTOPOMGR% synonym.

```
   SYN %AOFOPTOPOMGR%  = '&domain.TPO';
```

You should not include any backup operators in this synonym.

It is recommended to use system symbols for the focal point, backup, and intermediate focal point specification. In this case, you can update AOFOPFFP and AOFMSGSY accordingly and make it available in a general data set to all your systems, focal points, and targets. This avoids the same specification of two members on any single system.

You will need one set of autotasks for your primary focal point and a second set for your backup focal point. If you are using intermediate focal points, you will also need a set of operators for each of those (but only on the target systems that are defined to the IFP). Note that even in an IFP situation, the focal point will contact all target systems directly to obtain status and configuration data. The IFP is only used for alert forwarding.

Operator Profiles

This concerns statements in DSIOPF, which associate operator ids with logon profiles and the profiles themselves, which are defined in the DSIPRF concatenation.

Each operator who will be an NMC Administrator must be assigned a NetView logon profile which includes the NGMFADMN=YES key/value pair on its AUTH tag.

Each NMC user who needs to issue commands against resources through the NMC interface needs to be linked to a profile with the NGMFCMDS=YES key/value pair on its AUTH tag.

DSIPARM.DSICRTTD

The focal points need to be identified to your target systems. Uncomment and adapt the following line for any of your target systems:

```
*  DEFFOCPT PRIMARY=CNM02LUC,TYPE=ALERT,BACKUP=CNM99LUC
```

DSIPARM.DUIFPMEM

Uncomment and adapt the following 4 statements.

```
*USETCPIP = NO
*TCPANAME = &CNMTCPN
*SOCKETS  = 50
*PORT = 4020
```

Change USETCPIP to YES. Change the PORT number to an unused number in your system if necessary.

DSIPARM.DUIGINIT

Change the domain specification to your focal point domain.

DSIPARM.FLCSAINP

Copy this member to your user DSIPARM data set and rename it to FLCAINP.

DSIPARM.INGTOPOF

Define your sysplex to your NMC as described in “Step 26D: Customize the INGTOPOF File” on page 121.

Step 26C: Customize RODM

You need to configure RODM so that it will dynamically refresh the workstation when a number of fields other than DisplayResourceStatus is changed. To do this you need to ensure that certain RODM loader statements are processed whenever the GMFHS Data Model is reloaded.

Add the DD statement with member INGDYNRF in the NetView sample procedure EKGLOADP.

```
   ⋮
//*EKGIN1   DD DSN=&EKGIN1,DISP=SHR
//EKGIN1 DD DSN=&SQ1..V&NETVER..CNMSAMP(DUIFSTRC),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM1),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM2),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM3),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM4),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM5),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM6),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM7),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM8),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM9),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMA),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMB),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMC),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMD),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDME),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMF),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMZ),DISP=SHR
//*  Dynamic update of resources
//      DD DSN=&SQ2..V&SAMVER..SINGSAMP(INGDYNRF),DISP=SHR
//*
//*
   ⋮
```

Figure 23. Sample of RODM Load Procedure EKGLOADP

Step 26D: Customize the INGTOPOF File

The following overview of the operation mode of the SA OS/390 topology manager supplies some background for discussing the INGTOPOF file. Some familiarity with the class structure of RODM and with the BLDVIEWS tool is assumed.

During initialization, the SA OS/390 topology manager gathers information about generated SA OS/390 resources from the sysplex and stores the resources within RODM, prefixing their names with the current sysplex name. Usually not only the resources, but also the dependencies and major/minor relationships between resources will be represented in RODM (this depends on the OPTION statement in the INGTOPOF file, see Appendix B, "Syntax for INGTOPOF File", on page 167).

The INGTOPOF file supplies the SA OS/390 topology manager with the following information:

- which sysplexes there are and which of their member systems contain a SA OS/390 topology agent.
- the names of the data sets (members) that contain the definitions of the views.
- when views must be rebuilt during runtime, it is desirable that only those views be rebuilt to which new members have been added.

You will need to prepare the INGTOPOF input file. This contains information about the target domains and how they are grouped into sysplexes along with some additional information that affects the resources that are dynamically created.

Step 26: Installing and Customizing the SA OS/390 Topology Manager

The INGTOPOF file contains configuration information for the SA OS/390 topology manager. It must reside in DSIPARM. The records of the file consist of a keyword with one or more parameters. Comment lines must have an asterisk (*) in the first column. A '+' at the end of a line indicates that the record is continued in the next line.

The information is passed from the INGTOPOF file to the SA OS/390 topology manager with the help of the following keywords:

- SYSPLEX
- PROCOPS
- BLDVIEWS
- [LOCATION]
- [ANCHOR]
- [OPTION]
- [TEMPLATE]

The syntax of the statements in the INGTOPOF file is described in Appendix B, "Syntax for INGTOPOF File", on page 167.

A sample of INGTOPOF is provided in the SINGNPRM library.

To start the MultiSystem Manager and load the INGTOPOF file, use the MultiSystem Manager start command FLCAINIT.

Step 26E: Prepare BLDVIEWS Cards

You need to provide files with BLDVIEWS cards. These are required for the SA OS/390 resources to appear on the NMC workstation. These files will become part of the BLDVIEWS statement in the INGTOPOF file. The BLDVIEWS statement in the INGTOPOF file is used by the SA OS/390 topology manager to pass information to the BLDVIEWS tool which it invokes to produce the views of the objects. The BLDVIEWS tool writes information about views into RODM. The SA OS/390 topology manager is automatically invoked whenever you start SA OS/390 or you can invoke it with the INGTOPO command whenever you changed information in the INGTOPOF file or in the files with the BLDVIEWS cards.

To run the BLDVIEWS tool, use one of the following methods:

- via the SA OS/390 topology manager which invokes the tool
- via an external invocation of this tool (as a NetView command in a NetView session)

For information about the BLDVIEWS cards syntax refer to the appropriate NetView documentation.

The following three SA OS/390 BLDVIEW samples are provided in the SINGNPRM library matching the INGTOPOF sample file:

- INGBVIEW (sample view for SysOps objects)
- INGPVIEW (sample view for ProcOps objects)
- INGCVIEW (sample view for common objects)

Note: To start MultiSystem Manager and load the INGTOPOF file, use the MultiSystem Manager start command: `FLCAINIT`

Step 27: Copy and Update Sample Exits

Optional for:

SysOps, ProcOps, I/O Ops

Several sample exits are provided in the SINGSAMP library (for example, AOFEXSTA). You can use these samples to create your own exits. When used, they must be copied into a data set (either the enterprise-specific or domain-specific) in the DSICLD concatenation. These exits are called at fixed points during SA OS/390 processing. Therefore, you should look into each of the sample exits to determine whether you need to use and update it.

Updating and copying the sample exits allows you to add your specific processing. For more information on user exits, provided samples and advanced automation options, refer to *System Automation for OS/390 Customizing and Programming*.

Step 28: Install CICS Automation in CICS

This section describes the basic CICS Automation definitions that take place on CICS. Refer to the CICS documentation while performing these steps, especially the *CICS TS Resource Definition Guide*. These steps are performed on each CICS region.

Note: The TS queues EVEVCQUE and COLEEVEQ used by SA OS/390 CICS must not be defined as remote in your TST (temporary storage table).

Step 28A: SIT or Startup Overrides

On each CICS, ensure that the system initialization table (SIT) or startup overrides include the following:

```
PLTPI=xx,             where xx is the suffix to the startup PLT
PLTSD=yy,             where yy is the suffix to the shutdown PLT
MSGLVL=1,
BMS=(STANDARD|FULL)
```

Because CICS Automation maintains a long-running task in each CICS, review the AMXT, CMXT, and MXT values.

You may optionally add CN as your last startup override, whether from SYSIN or through the JCL. However, this is not necessary if you have added the &EHKVAR1 variable to the PARM of the CICS start command in the STARTUP item of the APPLICATION policy object. The following is an example:

```
MVS S cics,...,PARM='SYSIN,START=xxxx&EHKVAR1'
```

This is also the way the start commands are predefined in the sample databases.

Step 28B: Program List Table Definitions

Add the TYPE=ENTRY definitions shown in the following example to the post initialization program list table (PLT) for each CICS after the entry for DFHDELIM (as in phase 2).

```
DFHPLT TYPE=INITIAL,SUFFIX=xx
DFHPLT TYPE=ENTRY,PROGRAM=EVESTIEX
DFHPLT TYPE=ENTRY,PROGRAM=EVESTISP
DFHPLT TYPE=FINAL
```

Add the TYPE=ENTRY definitions shown in the following example to the shut down program list table (PLT) for each CICS.

```
DFHPLT TYPE=INITIAL,SUFFIX=yy
DFHPLT TYPE=ENTRY,PROGRAM=EVESPLTT
DFHPLT TYPE=ENTRY,PROGRAM=EVESYLMQ
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=FINAL
```

Assemble the PLT tables.

If you must define a new table, add the following definitions to the CSD:

```
DEFINE PROGRAM(DFHPLTxx) LANGUAGE(ASSEMBLER) GROUP(yourgroup)
DEFINE PROGRAM(DFHPLTyy) LANGUAGE(ASSEMBLER) GROUP(yourgroup)
```

Step 28C: Define Consoles

CICS Automation uses EMCS consoles to issue Modify CICS commands when managing CICS. Console definitions are required for correct CICS Automation operation.

Define consoles for autotasks to enable CICS Automation functions.

In an EMCS environment the autotask console names are determined, in order of precedence as follows:

1. If you are using AOCGETCN (that is, using the profiles shipped with the product) the name is determined by AOFCNMASK. For more information, see *System Automation for OS/390 Customizing and Programming* or *System Automation for OS/390 Defining Automation Policy*.
2. The CONSNAME parameter on the PROFILE statement in the task profile determines the EMCS console name. For more information, see *Tivoli NetView for OS/390 Administration Reference* and *Tivoli NetView for OS/390 Security Reference*.
3. By default the autotask name is used for the EMCS console name.

The SHUTOPER automated function of the base determines the autotask name used for shutdown; the autotasks defined for the CICS automated functions (see *System Automation for OS/390 Defining Automation Policy*) are used for broadcast. The Work autotasks AUTWRK*xx* also need to be defined to CICS also.

In addition, a console has to be defined for each NetView operator that may wish to inquire or control a CICS region.

This can be simplified by specification of the CICS Console Auto-install function for CICS Transaction Server V1.3 and later releases. Further information on this function can be found in the CICS Installation/Customization manuals.

For pre-Transaction Server V1.3 CICS systems, the following two PTFs provide a mechanism to define a pool of consoles: PQ09813 for CICS TS V1R1 and V1R2.

Step 28D: Transaction and Program Definitions

This step describes how to define the standard CICS Automation transactions and programs to CICS. To this purpose, the DFHCSDUP program is used.

The members required to run these jobs are provided with CICS Automation. However, some modifications are required, as described below:

> **Hint**
>
> You might want to backup your CSD's before doing this step.

For each CSD, run the EVESJ015 sample job. This job defines transactions and programs for CICS Automation in three groups: EVEGRP1, EVEGRP2, and EVEGRP3. Before you run it, modify the job as follows:

```
//STEPLIB  DD DISP=SHR,DSN=CICS.xxy.LOADLIB           (See note 1.)
//         DD DISP=SHR,DSN=ING.SINGMODC               (See note 2.)
//DFHCSD   DD DISP=SHR,DSN=yourCSD.DFHCSD             (See note 3.)
//SYSUT1   DD UNIT=SYSALLDA,SPACE=(1024,(100,100))
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
   UPGRADE  USING(EVECSDxx)                            (See note 4.)
   ADD GROUP(EVEGRP1) LIST(yourlist)                   (See note 5.)
   ADD GROUP(EVEGRP2) LIST(yourlist)
   ADD GROUP(EVEGRP3) LIST(yourlist)
/*
//
```

Notes:

1. Change the data set name of this file to match your CICS load library.
2. If required, change the data set name on this file to match your CICS Automation load library.
3. Change the data set name to match your CICS CSD for this CICS.
4. Change *xx* to 51, 52, 53, 62, or TS , depending on your CICS level.

   51 refers to CICS TS for SA OS/390 R1

   52 refers to CICS TS for SA OS/390 R2

   53 refers to CICS TS for SA OS/390 R3

   62 refers to CICS TS for SA OS/390 V2R2

   TS refers to all CICS TS for z/OS versions currently supported
5. Change your list for EVEGRP1, EVEGRP2, and EVEGRP3 to the name of the group lists for this CICS.

Step 28E: DFHRPL and the CICS Automation Library

Add the ING.SINGMODC data set library to the DFHRPL concatenation.

Step 29: Install IMS Automation in IMS

Step 29A: Modify and Run the IMS SYSGEN

DBCTL regions do not require IMS sysgen information. Therefore this step can be omitted.

1. Add the statements in ING.SINGSAMP member EVISI002 into your IMS gen for each IMS.

   ```
   APPLCTN  PSB=EVISPPII,
     PGMTYPE=(BATCH,,1),
     SCHDTYP=SERIAL
    TRANSACT CODE=EVITPPI1, PPI TRANSMISSION TRAN
     MSGTYPE=(MULTSEG,NONRESPONSE),
     SCHD=1,
     INQ=(YES,NORECOV),
   ```

```
SEGSIZE=0,
SEGNO=0,
MODE=SNGL,
SERIAL=YES
```

2. Make sure that the APPLID parameter on the COMM macro in the GEN contains the appropriate applID. The APPLID parameter on the COMM macro for IMS must be specified. This parameter must match the one specified in the APPLID parameter statement in the policy database, when defining a particular IMS control region. If the APPLID is left to default in the IMS GEN, the MVS jobstep name will be used for the APPLID name for this IMS, thus causing the IMS Automation PPI to fail to initialize. A mismatch will occur at initialization time.
3. This substep is different for different versions of IMS. Place the DFSAOUEO module in an APF-authorized library included in STEPLIB for the IMS region. This could be the RESLIB, or another library in the concatenation.
4. Perform a GEN for each IMS.

Step 29B: Define IMS PSB Entries

1. Merge the statements in ING.SINGSAMP member EVISI001 with your existing PSB gen for each IMS, or manually add the following to it:

```
    PCB TYPE=TP,MODIFY=YES,EXPRESS=YES
    PSBGEN
LANG=ASSEM,PSBNAME=EVISPPII,CMPAT=YES,IOASIZE=1024
   END
```

2. Run the PSB and ACB gens for each IMS.

Note: The output of the ACBGEN utility for the PSB ″name″ will indicate how much space is required in the CSA PSB pool.

Step 29C: Define IMS Security Gen Entries

Add the statements in ING.SINGSAMP member EVISI003 to your security maintenance utility input and run a security gen for each IMS, or code them as shown below. This will give transaction EVITPPII access to all commands.

```
)( CTRANS   EVITPPI1    /* GENERATE TRANSACTION SECURITY */
   TCOMMAND *           /* ALL COMMANDS */
```

For users of IMS Version 6 or above, any NetView operator that issues 'Display Shutdown Status' (option 1.4 of the SA OS/390 IMS operator command dialogs) must have their NetView terminal authorized to issue IMS display commands, otherwise an IMS DFS093 error message will be issued. The Display Shutdown Status facility is not supported for IMS versions below version 6.

If you plan to use the Display Shutdown Status facility to display shutdown status information for IMS systems running on remote domains, then you should review the entries in the SEND COMMAND OPERS policy item of the ENTERPRISE object (see *System Automation for OS/390 Defining Automation Policy*). This is because NetView command RMTCMD is used to route the necessary IMS display status commands to remote domains. RMTCMD runs on the remote domain using the current userID on the local domain unless this userID is modified using the SEND COMMAND OPERS policy item.

Note: Security checks will be performed in NetView before IMS Automation is invoked.

Step 29D: Define IMS BMP Procedure

The BMP handles communications between the IMS AOIEXIT and the NetView program-to-program interface. The BMP is initialized as a started procedure requested by EVInAOIX (n=IMS version number). Its start is unconditional and not modified by any entries in the policy database. To customize the BMP, perform the following steps:

1. Copy ING.SINGSAMP member EVISI004 to your user PROCLIB as the name PPI*imsid*, where *imsid* is the one-to-four-character IMS ID of the IMS control region. If you need to change the name of the procedure to start the BMP, you must also change it in the corresponding IMS program-to-program interface initialization member, EVISPINM.
2. Update the STEPLIB statement to point to the IMS RESLIB and SEVIMODx you are using.
3. Update the PROCLIB statement to point to the PROCLIB you are using.
4. Update the imsID variable to the IMS ID of the control region you are using.

Step 29E: Create PPI Initialization Member for IMS

This initialization member, EVISPINM, defines what IMS messages to send to NetView and defines PPI control information.

Note: You can use a single table for every IMS or, optionally, you can have a separate table for each IMS. The following instructions assume you will use a single table. To create separate tables, see ″Optional Additions to the PPI″ in topic 2.1.4.

Perform the following steps:

1. Use the EVISJ020 sample JCL to assemble your program-to-program interface initialization member, EVISPINM.
2. Place the assembled member into a library accessible by both the control region and the BMP procedure created in “Step 29D: Define IMS BMP Procedure”, typically IMS RESLIB.

Step 29F: Specify CMDMCS parameter

Modify all IMS Control region and IMS DB control region JCL to specify the following parameter:

CMDMCS=Y
This is required for correct operation of IMS product automation.

Step 30: Install OPC/TWS Automation in OPC/TWS

Step 30A: Add Libraries to OPC/TWS

Add your SINGMOD1 library and the NetView CNMLINK library containing CNMNETV to the OPC steplib. Alternately, you may add these libraries to LINKLST. You should have already APF authorized these libraries.

Step 30B: Update OPC/TWS parameters and exits.

A recycle of OPC is required for installing the exit 7 module EQQUX007. If you are using an existing exit 7, you can combine this exit with OPC Automation-supplied modules.

Step 30: Install OPC/TWS Automation in OPC/TWS

OPC Automation supplies EQQUX007 to detect workstations used for NetView communication. The following modules are used as part of that process:

```
EQQUX007
UX007001
UX007002
UX007003
```

EQQUX007 is the exit driver program. It calls other modules in turn, as if OPC is calling each module directly. The driver searches for UX007001 through UX007010. UX007001 through UX007003 are supplied with OPC Automation. If you have an existing exit 7, rename your module from EQQUX007 to UX007004.

The called routines are passed the same parameters the call to EQQUX007.

If you wish to add additional exit 7 modules, then use the next available name, such as UX007005. This makes it easier to integrate exits supplied by various products. Also, since modules are loaded dynamically by the exit driver on each invocation, you may add, delete, or modify an exit module without recycling OPC.

You must specify the CALL07(YES) parameter in the OPC/ESA initialization parameters.

Other initialization parameters must be specified in the OPC initialization member (EQQPARM) so that OPC will issue some of its messages to the MVS console.

The DURATION, ERROROPER, LATEOPER, and OPCERROR messages are automated by OPC Automation. The RESCONT and QLIMEXCEED messages are useful for further customer automation.

You must specify the following in EQQPARM:

```
ALERTS WTO (DURATION
   ERROROPER
   LATEOPER
   RESCONT
   OPCERROR
   QLIMEXCEED)
```

In addition, you must edit the OPC-supplied message members for certain messages.

The following messages are automated and may require changes to the OPC or TWS supplied message members in the SEQQMSG0 data set .

```
Message   Member
     -------- ---------
     EQQW065I  EQQW06
     EQQW011I  EQQW01
     EQQN013I  EQQN01
     EQQZ086I  EQQZ08
     EQQE026I  EQQE02
     EQQE036I  EQQE03
     EQQZ128I  EQQZ12
     EQQZ201I  EQQZ20
```

Modify these message members to include WTO=YES for the indicated message IDs. Full details for customizing OPC can be found in *Tivoli Workload Scheduler for z/OS Customization and Tuning*.

Chapter 9. Installing SA OS/390 on Workstations

This chapter contains information on how to install those parts of SA OS/390 that are required on workstations:

- "Installing the NMC Workstation"
- "Installing and Customizing the TEC Event Server Workstation" on page 135

The workstation components can be installed on any workstation that meets the requirements listed in Chapter 1, "SA OS/390 Prerequisites and Supported Equipment", on page 3. One or more workstations can be installed for users to monitor and control the systems being managed with SA OS/390.

The code for the SA OS/390 NMC exploitation is supplied with the host code that is installed using SMP/E. Installing the SA OS/390 NMC exploitation will enable you to issue the most important SA OS/390 processor operations and system operations commands from all NMC workstations.

Note: The NMC installation as described in "Installing the NMC Workstation" is performed on the NMC Server and the NMC clients. After this installation, you need to restart the individual NMC clients.

Installing the NMC Workstation

If you already have an NMC environment installed, you can continue with the actions described in the remainder of this section. Having completed these, you can use the SA OS/390 NMC exploitation as described in *System Automation for OS/390 User's Guide*. This will enable you to issue a selection of SA OS/390 processor operations and system operations commands from all NMC workstations.

The following packed files for the SA OS/390 NMC exploitation are available after your SMP/E installation:

- ING.SINGPWS1(INGNMCZP): packed file for Windows or OS/2 Workstations; download this file with extension ZIP and unpack with an appropriate tool (WINZIP or PKZIP).
- ING.SINGPWS1(INGNMCTZ): packed file for UNIX® Workstations; download this file with extension TAR.Z and unpack and uncompress with an appropriate tool (*uncompress* and *tar*).

The content of each packed file is divided into a support for system operations commands and a support for processor operations commands. Both packages include two NMC response files. One response file contains the system operations commands, the other one contains the processor operations commands. The response files include the definitions and profiles for

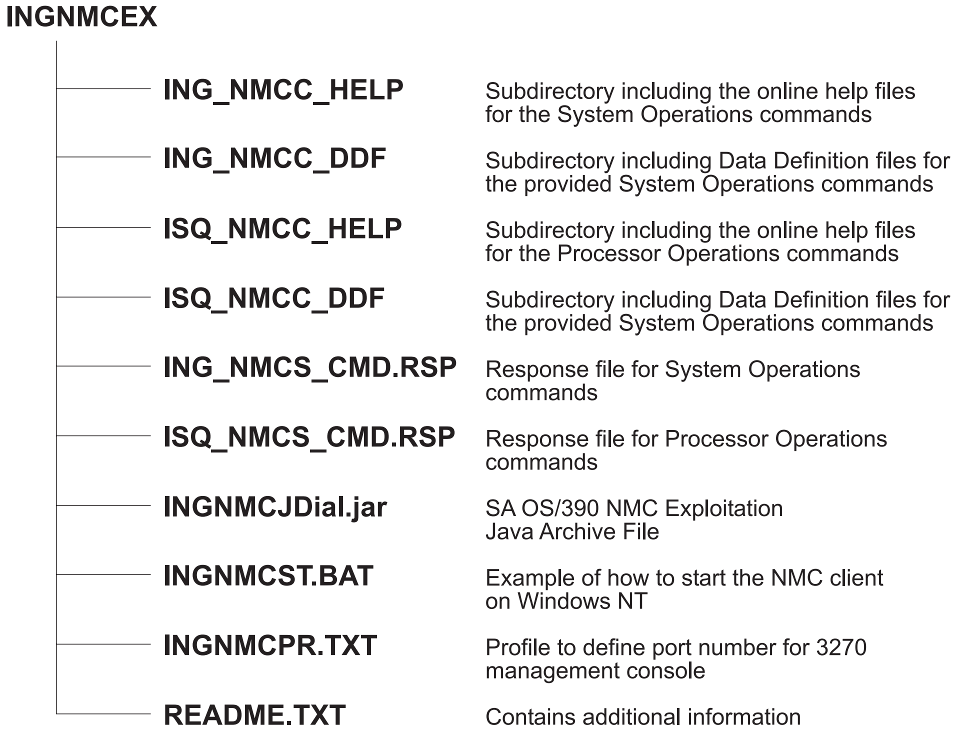
ING_SO_OPER
SystemOperation Operator

ING_PO_OPER
ProcessorOperation Operator

ING_SA_OPER
SystemAutomation Operator (definition for both the system operations and processor operations commands)

Furthermore there are two subdirectories for the related data definition files and two subdirectories with the online help in HTML format.

With this separation of system operations and processor operations commands you may install either the system operations commands or the processor operations commands or both depending on your needs. The installation has to be done manually, as there is no common installation tool for the several supported platforms. This requires that you are familiar with the common commands of your workstation operating system.



INGNMCEX

ING_NMCC_HELP	Subdirectory including the online help files for the System Operations commands
ING_NMCC_DDF	Subdirectory including Data Definition files for the provided System Operations commands
ISQ_NMCC_HELP	Subdirectory including the online help files for the Processor Operations commands
ISQ_NMCC_DDF	Subdirectory including Data Definition files for the provided System Operations commands
ING_NMCS_CMD.RSP	Response file for System Operations commands
ISQ_NMCS_CMD.RSP	Response file for Processor Operations commands
INGNMCJDial.jar	SA OS/390 NMC Exploitation Java Archive File
INGNMCST.BAT	Example of how to start the NMC client on Windows NT
INGNMCPR.TXT	Profile to define port number for 3270 management console
README.TXT	Contains additional information

Figure 24. Directory Structure of Unpacked Files

Installation Steps on the NMC Server

Perform the following steps to install SA OS/390 NMC exploitation on the NMC Server (it should be noted, the term UNIX in the following steps refers to all forms of UNIX derivatives, including AIX and SUN SOLARIS):

1. Download the appropriate packed file in binary format to the NMC Server.
2. Unpack the file into a temporary directory of the NMC Server, using an appropriate tool for the NMC Server operating system. You will obtain the directory structure for the unpacked files as shown in Figure 24.

3. Copy the required help files as follows:

Environment	From Directory	To Your Directory
WIN or OS/2	*tmp*\INGNMCEX\ING_NMCC_HELP and/or *tmp*\INGNMCEX\ISQ_NMCC_HELP	[BINDIR]\TDS\server\db\current\help
UNIX	*tmp*/INGNMCEX/ING_NMCC_HELP and/or *tmp*/INGNMCEX/ISQ_NMCC_HELP	$BINDIR/TDS/server/db/current/help

where *tmp* stands for the directory where you downloaded the files.

Note: ***BINDIR*** is an environment variable set by your NMC installation and indicates that this is a subdirectory of your installed NMC product. For example:

```
usr\local\Tivoli\bin\w32-ix86\
```

4. Copy the required data definition files as follows:

Environment	From Directory	To Your Directory
WIN or OS/2	*tmp*\INGNMCEX\ING_NMCC_DDF and/or *tmp*\INGNMCEX\ISQ_NMCC_DDF	[BINDIR]\TDS\server\config\ddf\c
UNIX	*tmp*/INGNMCEX/ING_NMCC_DDF and/or *tmp*/INGNMCEX/ISQ_NMCC_DDF	$BINDIR/TDS/server/config/ddf/c

5. Copy the required response files from INGNMCEX as follows:

Environment	To Your Directory
WIN or OS/2	[BINDIR]\TDS\server\sample
UNIX	$BINDIR/TDS/server/sample

6. Copy the Java archive file INGNMCJDial.jar from INGNMCEX as follows:

Environment	From Directory	To Your Directory
WIN or OS/2	*tmp*\INGNMCEX	[BINDIR]\TDS\server\db\current\lib
UNIX	*tmp*/INGNMCEX	$BINDIR/TDS/server/db/current/lib

7. Verify the following:
   a. To operate the NMC Server you must be logged on to NetView via a 3270 host session.
   b. Your NetView user ID must have NGMF administrator rights.
   c. The NMC Server must be started and active.
   d. The connection from the NMC Server to NetView must be established.
8. Start the *Command Profile Editor batch utility* (CPEBATCH) with:
   a. for WIN or OS/2 environment
      - `[BINDIR]\TDS\server\sample\ING_NMCS_CMD.RSP` and/or
      - `[BINDIR]\TDS\server\sample\ISQ_NMCS_CMD.RSP`

      and the -i and -g parameters
   b. for UNIX environment

- `$BINDIR/TDS/server/sample/ING_NMCS_CMD.RSP` and/or
- `$BINDIR/TDS/server/sample/ISQ_NMCS_CMD.RSP`

and the -i and -g parameters

With this step, you load the delivered commands into the NetView internal database. For information on how to use this batch utility, refer to *NetView Management Console User's Guide*. For a detailed description of how to maintain and manipulate response files for the NMC topology server, go to the SA OS/390 Web page at `http://www.ibm.com/servers/eserver/zseries/software/sa/adds/hint03.html`.

9. Start the CPE online utility to apply the new profiles to the individual operators. You can also use this online utility to add or remove single commands to or from your individual profile. For details of the CPE see *NetView Management Console User's Guide*.

   Note: The CPE online utility is available for INTEL platforms only. For other platforms you need to edit the files with a text editor. For more information refer to *NetView Management Console User's Guide*.

Installation Steps on the NMC Client

You must have the ***NetView 3270 Management Console*** installed. Refer to the *NetView Management Console User's Guide* for information on how to do this.

1. Set the environment variable TCONSOLE_CLASSPATH:
   a. for WIN or OS/2 environments pointing to:
   `[NMC_Client_Installation_path]\TDS\client\lib\INGNMCJDial.jar`
   b. for UNIX environments pointing to:
   `[NMC_Client_Installation_path]/TDS/client/lib/INGNMCJDial.jar`

   Refer to Figure 25 on page 134 for a sample batch file.
2. On the individual NMC Clients: Restart your NetView Management Console to incorporate your changes.
3. Customize the NetView 3270 Management Console:
   a. On the NMC, select an SA OS/390 resource from an existing view. For this resource, select an SA OS/390 command that needs to be transferred to the NetView 3270 Management Console, for example, the INGVOTE command. Click on INGVOTE to display the NetView 3270 Management Console, that does not show any output yet.
   b. Select ***Session Services*** from the NMC menu bar, and choose ***Add/Delete/Modify Session*** from the menu items. This opens the ***Add/Delete/Modify Session*** window.
   c. In the ***Full Screen Session Name*** field of this window type: `SA`
   d. In the ***Start command String*** field type, for example: `window date`
   (You can enter any valid NetView command.)
   e. Select the radio button ***Immediate***
   f. From the ***Session Options*** select: ***Start Automatically***
   g. Press the ***Add*** push button, then the ***Save*** push button to save your changes
   h. Press the ***Done*** push button to exit this window
   i. In the NMC, select the added *SA* pull-down choice from the ***Session Services*** menu bar item

j. To verify the customization, issue the INGVOTE command to display the desired output

Sample to Start the NMC (for Windows NT Environment)

```
@rem **************************************************************************
@rem IBM System Automation for OS/390 NetView Management Console Exploitation
@rem Sample Program - 5645-006
@rem               (C) Copyright IBM Corp. 2002
@rem                    All rights reserved.
@rem
@rem SAMPLE PROGRAM - NO WARRANTY EXPRESSED OR IMPLIED
@rem
@rem You are hereby licensed to use, reproduce, and distribute these sample
@rem programs as your needs require.  IBM does not warrant the suitability or
@rem integrity of these sample programs and accepts no responsibility for their
@rem use for your applications.  If you choose to copy and redistribute
@rem significant portions of these sample programs, you should preface such
@rem copies with this copyright notice.
@rem **************************************************************************
@rem
@rem PRODUCT          (System Automation for OS/390)
@rem COMPONENT        (NMC Exploitation)
@rem FIRST_RELEASE    (V2R1)
@rem LAST_CHANGE      (11Jan2002)
@rem
@rem MODULE_NAME      (ingnmcst.bat)
@rem DESCRIPTIVE_NAME (Start the NMC Topology Console)
@rem **************************************************************************
@rem
@rem Function:  This sample shows how the NMC Topology Console can be
@rem            started. This sample was written for the Windows NT environment
@rem            and NMC 1.3.0.1.
@rem
@rem Usage:
@rem
@rem - The following is a sample which will NOT properly work until customer
@rem   installation specific data is provided.
@rem
@rem - Adapt the drive and path statements to reflect your installation
@rem   environment.
@rem   This example assumes that the NMC Topology Console was installed on
@rem   drive E:.
@rem
@rem - A good location to put this file is the directory:
@rem   E:\usr\local\Tivoli\bin\generic_unix\TDS\client\bin
@rem   If it is necessary it can be stored anywhere else.
@rem
@rem - Call this file from a icon on your desktop or from Windows
@rem   Start-Programs-Netview-... pull-down or from the command line.
@rem **************************************************************************

@setlocal

@rem Changes the user's current working directory to the 'bin' directory in
@rem the "base" console installation path.
E:
cd E:\usr\local\Tivoli\bin\generic_unix\TDS\client\bin

@set TIVOLI=e:\usr\local\Tivoli\bin\generic_unix\Tds
@set INGJAR=\client\lib\INGNMCJDial.jar
@set FLBJAR=\ibmflb\jars\tivflb13.jar

set TCONSOLE_CLASSPATH=%TIVOLI%%FLBJAR%;%TIVOLI%%INGJAR%
tconsoleNT.bat   .. -key nmc
@endlocal
```

Figure 25. Sample to Start the NMC (for WIN Environment)

Installing and Customizing the TEC Event Server Workstation

The TEC event server can either run on a UNIX workstation or on a Windows NT® workstation. The following example describes the installation on UNIX. For the Windows NT installation, please use Windows NT command syntax.

1. Download the package file INGPTEC containing the workstation code from the host system to your workstation as a binary file. To download the package, you can, for example, use *FTP*. Choose as target path name any directory where you want to store the tarfile temporarily and unpack it for installation.

   Using FTP, the command is for example:

   ftp *<hostname>*

   You will be prompted for your user ID and password. After logon to your OS/390 system, type:

   binary

   get *<dataset>*(INGPTEC) <PATH>/satec.tar

   quit
2. On your workstation, type:

   cd *<PATH>*
3. Unpack the package file `<PATH>/satec.tar`.

   tar -xvf <PATH>/satec.tar

   This will unpack the workstation code for subsequent installation into the current directory (*<PATH>*).

   On Windows NT, you can find the `tar` command in

   `c:\tivoli\bin\w32-ix86\tools\tar.exe`
4. Install the appropriate Tivoli install package
   a. From the Tivoli desktop select ***Install->Install Product*** and follow the Install Product dialog
   b. Set the media path to the *<PATH>* which contains the SA OS/390 specific install packages.
   c. Select the product to be installed.
   d. Close the Install Product dialog after installation.
5. Verify the installation. The files listed in Table 20 should be stored in the correct directories.

Note:

After installation, the binary files are stored in the following directory:

`$BINDIR/SAOS390/NotificationService`

The environment variable ***BINDIR*** is set when installing the Tivoli Framework. As default, it points to

`/usr/local/Tivoli/bin/$INTERP`

The environment variable ***INTERP*** denotes the platform where Tivoli is used, and can be for example *aix4-r1*.

Table 20. Notification Service Product Workstation Files

Member Name	Type	Purpose
tecad_sa390msg.baroc	TEC *baroc* file	defines event classes
tecad_sa390msg.rls	TEC *rls* file	defines rules
nvcons.ksh	korn shell script (*ksh*)	starts NetView 3270 Management Console for UNIX

Table 20. Notification Service Product Workstation Files (continued)

Member Name	Type	Purpose
nvcons.bat	batch file	starts NetView 3270 Management Console for Windows NT
tecad_sa390msg.tll	task library definition	contains the task library

If the NetView 3270 Management Console is not yet installed on your workstation, you can download it from the internet:

http://www.tivoli.com/z_nv390/javaclient.html

Activating the Installed Files

You need to activate the following files of type:
- *rls* files
- *baroc* files
- *tll* files

Loading Classes and Rules

After downloading the files on the Tivoli workstation, several files are available in the directory

```
$BINDIR/SAOS390/NotificationService
```

The following instructions describe the steps required to activate the installed files at the TEC event server. These steps are necessary in order to exploit the GEM Event/Automation Service for sending SA OS/390 events to TEC. See the *Tivoli Enterprise Console User's Guide Volume II* for a detailed description of the following commands:

- Use an existing rule base with GEM classes imported.
- Import the class file (.baroc) into the rule base:

  wimprbclass tecad_sa390msg.baroc <rbname>
- Import the rules file (.rls) into the rule base:

  wimprbrules tecad_sa390msg.rls <rbname>
- Compile the rule base:

  wcomprules <rbname>
- Load the rule base into the TEC event server:

  wloadrb -u <rbname>
- Stop the TEC event server:

  wstopesvr -d
- Start the TEC event server:

  wstartesvr

Creating the System Automation Task Library

If the NetView 3270 Management Console is installed on a workstation in your network managed by Tivoli, you can use a task provided in the System Automation Task Library to start the NetView Client from the Tivoli Enterprise Console.

Depending on the platform where the NetView 3270 Management Console is installed, either a shell script (`nvcons.ksh`) for UNIX platforms or a batch file (`nvcons.bat`) for the Windows NT platform needs to be modified.

Note:

Windows NT setup:

- Modify the PATH variable via ***System Setup*** and add the path where ***Java*** is installed.
- Modify ***Tivoli Object Dispatcher Service*** via ***System Setup*** and allow the service to interact with the desktop in order to display the NetView Console.

- Edit the according file as appropriate:

  CLASSPATH variable (UNIX and Windows NT)
  Set to the path where java classes are installed, add NetView class directory (see NetView documentation for details).

  NV variable (UNIX only)
  Set to path where NetView 3270 Management Console is installed on your system.

  Java variable (UNIX only)
  Set to path where Java is installed on your system.

  CD <NetViewDir> (Windows NT only)
  Set to directory where the NetView 3270 Management Console is installed on your system.

  Example for UNIX (nvcons.ksh):

```
export NV=/IBMFLB
export JAVA=/develop_driver/java/Java
export CLASSPATH=$JAVA/classes:$NV:$NV/sguide:$NV/sguide/SGJ023A.ZIP
:$NV/sguide/sguide.zip:$NV/src/ibmflb:$NV/jhelp
```

  Example for Windows NT (nvcons.bat):

```
set CLASSPATH=.;C:/users/java/lib/classes.zip
cd \IBMFLB
```

- Import the Task Library by using the wtll command (see the *Tivoli Management Platform Reference Guide* for details on this command).

```
wtll -p <policy region> -P <preprocessor> tecad_sa390msg.tll
```

  where

  <policy region>
  specifies the policy region in which to create the new task library. The policy region must exist within the local TMR.

  <preprocessor>
  specifies the path to the program to use as a preprocessor on the import file before it is parsed. The import file tecad_sa390msg.tll does not need to be preprocessed, so instead of specifying for example a C or C++ preprocessor, the command /bin/cat could be used.

Customization of the Tivoli Enterprise Console

Before you perform the steps described in this section, you should be familiar with the Tivoli terms ***event groups*** and ***event sources***. These are introduced in *Tivoli Enterprise Console User's Guide Volume I*.

In Tivoli, you may monitor events belonging to a group which may originate from a certain source or from different sources. In order to enable the TEC event server to handle the SA OS/390 specific events, you may need to define the appropriate source to TEC.

Customization of the Tivoli Enterprise Console

To enable Tivoli administrators to monitor events on their event consoles, you need to define one or more appropriate event groups (with events from the defined event sources) and assign these groups to the respective administrators' event consoles.

The following sections outline a sample sequence of definition steps you may perform to:

- define the appropriate ***event source*** to TEC
- define an event group
- assign the defined event group to an Tivoli administrator's Tivoli Enterprise Console

All steps described in this subchapter can be performed from the Tivoli desktop.

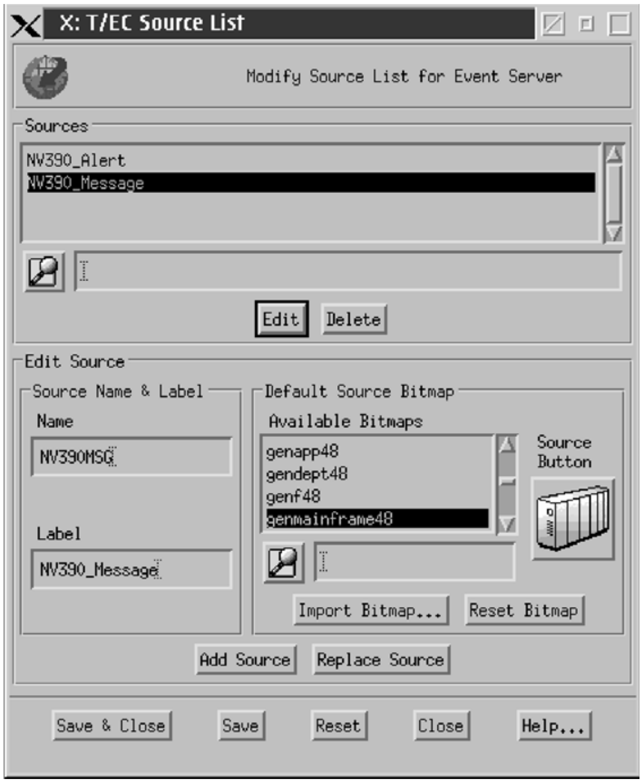
After you configured each event adapter you plan to use, the first step in the process of customizing the Tivoli Enterprise Console is to define ***event sources***. This is outlined in the following section. A Tivoli administrator called *tmeadm1* is used in the sample dialogs.

Defining Event Sources

The following sources are available after you installed Tivoli GEM Event/Automation Service and TEC Notification by SA OS/390 in your environment:

- *NV390MSG*
- *NV390ALT*

To define them to the Tivoli Enterprise Console select **Sources...** from the **Event Server** icon's pop-up menu to display the ***TEC Source List*** dialog as shown in Figure 26 on page 139).



Sources: NV390_Alert, NV390_Message

Name: NV390MSG

Label: NV390_Message

Available Bitmaps: genapp48, gendept48, genf48, genmainframe48

Figure 26. TEC Source List Dialog

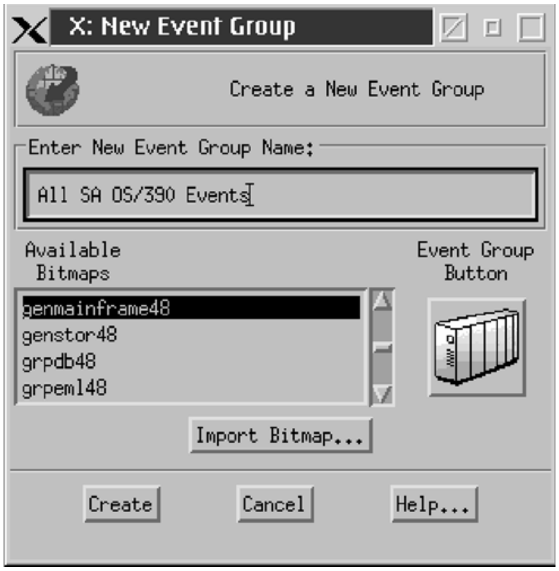
In this dialog you enter one of the names of the available sources into the ***Name*** entry field and a description of the source into the ***Label*** entry field. You can also assign an icon that will represent the source on the Tivoli desktop. For details see *Tivoli Enterprise Console User's Guide Volume I*.

Creating and Defining Event Groups

TEC uses event groups to direct events to the appropriate Tivoli administrator's TEC event console. You must configure a set of filters to determine which incoming events are included in an event group.

To create an event group select **Event Groups...** from the **Event Server** icon's pop-up menu to display the ***Event Group Management*** dialog.

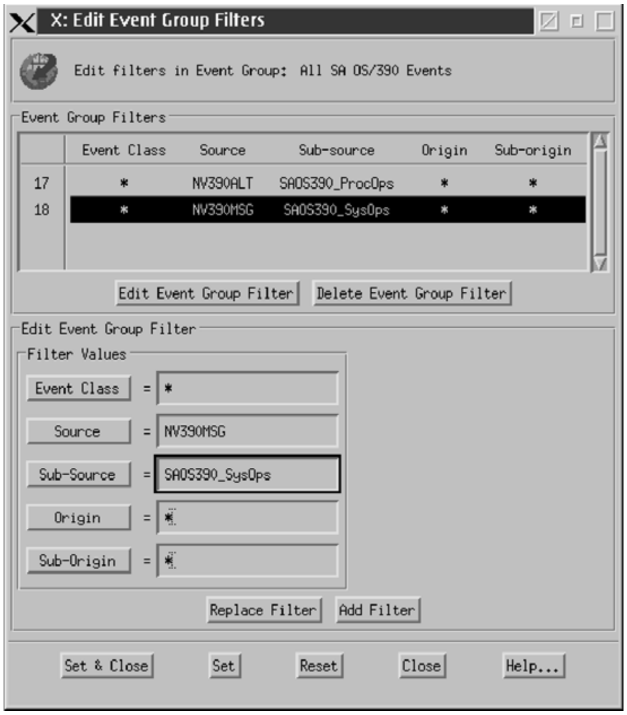
Then select **New...** from the **Event Group** menu to display the ***New Event Group*** dialog as shown in Figure 27 on page 140.



Enter New Event Group Name: All SA OS/390 Events

Figure 27. New Event Group Dialog

In this dialog you enter a name for the event group into the ***Enter New Event Group Name*** entry field, like **All SA OS/390 Events** in our example. You can also assign an icon that will represent the event group on the Tivoli desktop. For more details see *Tivoli Enterprise Console User's Guide Volume I*. After having assigned an icon to the event group, press the ***Create*** push button to create the event group **All SA OS/390 Events** and display the ***Edit Event Group Filters*** dialog as shown in Figure 28 to specify filter criteria for events that you may want to belong to this event group.



Edit filters in Event Group: All SA OS/390 Events

	Event Class	Source	Sub-source	Origin	Sub-origin
17	*	NV390ALT	SAOS390_ProcOps	*	*
18	*	NV390MSG	SAOS390_SysOps	*	*

Event Class = *

Source = NV390MSG

Sub-Source = SAOS390_SysOps

Origin = *

Sub-Origin = *

Figure 28. Edit Event Group Filters Dialog

An event group filter defines the events that are considered to be member of an event group. To create the filter for an event group, define the filter by setting the desired values for the desired attributes.

- All events originating from SA OS/390 messages have *NV390MSG* as source and *SAOS390_SysOps* as subsource.
- All events originating from SA OS/390 alerts have *NV390ALT* as source and *SAOS390_ProcOps* as subsource.

You may want to make use of this when defining your event group filters. The sample shows the definition of an event group filter for all SA OS/390 related events.

Press the ***Add Filter*** push button when you have specified all of the filter criteria.

More details about filter criteria for event groups are in *Tivoli Enterprise Console User's Guide Volume I*.

Press the ***Set & Close*** push button to save the filter. The Tivoli Enterprise Console branches back to the ***Event Group Management*** dialog where you can select **Save** from the **Event Group** menu to save your changes.

Assign Event Groups to Event Consoles

The last step in customizing your Tivoli Enterprise Console is to assign event groups to the event consoles of certain Tivoli administrators. This ensures that a Tivoli administrator will only see events that fulfil the filter criteria of the event group that you want to assign to that special Tivoli administrator. To invoke the dialog ***Assign Event Groups*** that lets you achieve this, select **Assign Event Groups...** from the Tivoli administrator event console icon's pop-up menu (in our sample, the *tmeadm1* icon) to display the ***Assign Event Groups*** dialog as shown in Figure 29.

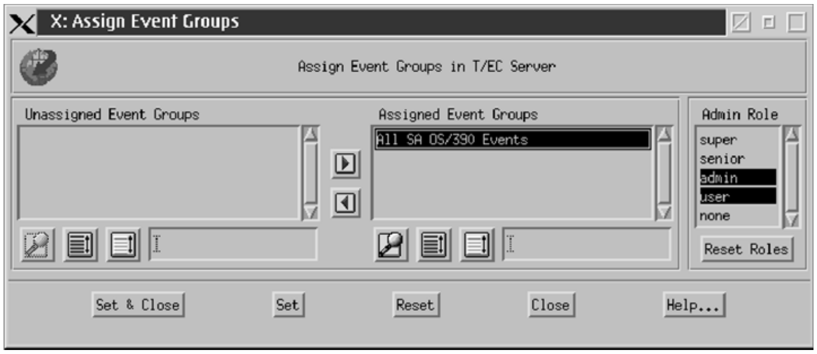
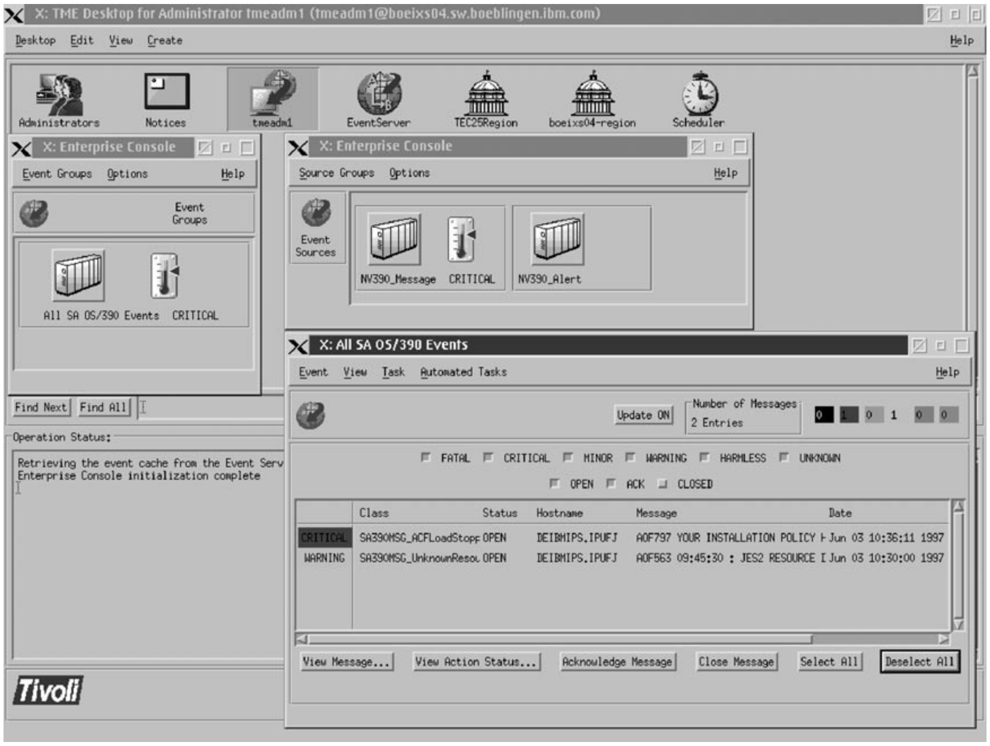


Figure 29. Assign Event Groups Dialog

The result of the customization could look similar to the Tivoli desktop shown in Figure 30 on page 142. This figure shows the desktop as it might be seen by our sample Tivoli administrator *tmeadm1*.



```
Retrieving the event cache from the Event Serv
Enterprise Console initialization complete
```

	Class	Status	Hostname	Message	Date
CRITICAL	SA390MSG_ACFLoadStopp	OPEN	DEIBMIPS.IPUFJ	AOF797 YOUR INSTALLATION POLICY H	Jun 03 10:36:11 1997
WARNING	SA390MSG_UnknownResou	OPEN	DEIBMIPS.IPUFJ	AOF563 09:45:30 : JES2 RESOURCE I	Jun 03 10:30:00 1997

Figure 30. Customized Tivoli desktop of Tivoli administrator tmeadm1

Part 3. Appendixes

Appendix A. Security and Authorization

This appendix describes how to install security options on your system.

Securing Focal Point Systems and Target Systems

Your operations staff and automation facilities at both focal point system and target systems need to be authorized to manage the resources in their environment. You can control human and automation operator authority through the password security provided by either:

- NetView
  - Operator definition file (DSIOPF)
  - Automation table
  - RODM access information
- A SAF-based security product such as RACF

The NetView facilities limit use of commands and keywords to authorized operators and limit an operator's span of control to specific systems. Access to the SA OS/390 graphic interface is controlled by user ID, password, and RODM access information. Logic in the NetView automation table or in your automation routines can verify the source of a message before taking an action.

RACF can be used to limit the use of OS/390 system commands to authorized operators.

When a target system is in the same sysplex as the focal point system, and your security product supports it, it is recommended that you share security definitions.

Operator Profiles

This sections provides information about operator profiles in two environments, in migrated ones and in RACF-based NetView ones.

Migrated Environments

Whether your operators work in a NetView environment that is RACF based or not RACF based, their authorization is:

```
AUTH CTL=GLOBAL
```

The following are all members of ING.SINGNPRF:

- AOFPRFAO
  - For all of system operations except the automatic restart management (ARM) element restart checker
- AOFPRFPI
  - For system operations automatic restart management element restart checker
- ISQPROF
  - For processor operations autotasks

Controlling Access to Console Commands for I/O Operations

The steps for doing this are

1. Define I/O console commands to RACF

2. Define consoles to RACF
3. Define TSO users who use the CONSOLE command to RACF
4. Allow consoles or TSO users to access I/O console commands

Use the procedure name you use to start I/O operations in the place of *ioproc*. This is either the name on the `EXEC` statement in the procedure, or if there is no name, the name of the member in `SYS1.PROCLIB` (or other procedure library that you use).

If you have already defined users and consoles to RACF, you need to do only the following:

```
RDEFINE OPERCMDS MVS.*.*.ioproc.* UACC(NONE)
PERMIT MVS.*.*.ioproc.** ID(userid) CLASS(OPERCMDS) ACC(UPDATE)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

Defining I/O Console Commands to RACF: The information that follows describes the steps needed to tell RACF about I/O commands entered from an OS/390 console.

All-or-Nothing Control: The following commands prevent any console user from controlling I/O operations:

```
SETROPTS GENERIC(OPERCMDS)
RDEFINE OPERCMDS MVS.*.*.ioproc.* UACC(NONE)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

These commands do the following:

```
SETROPTS GENERIC(OPERCMDS)
```

SETROPS

This command tells RACF to use generic names for the class of resources called `OPERCMDS` (for operator commands). This means that the "*" character can be used to replace any name, and "**" can replace any subsequent names including no name at all.

```
RDEFINE OPERCMDS MVS.*.*.ioproc.* UACC(NONE)
```

RDEFINE

This command defines to RACF the resource name. In this case the name is the one that is used to protect a certain command. The positional parameter, `OPERCMDS`, tells RACF what kind of resource the name is used for.

Next the name is supplied, using generics. This name follows the OS/390 resource naming convention. Using this name, OS/390 console services can check with RACF for users' authority to invoke commands through console interfaces. In this case, the first * means we are protecting *any* console command that can affect I/O. These are the START, STOP, MODIFY, CANCEL, and FORCE commands. OS/390 offers a further distinction between FORCE with and without the `ARM` parameter.

The second "*" is necessary because OS/390 allows you to protect either submitted jobs (using `JOB` as the third position in the resource name) or started tasks (using `STC`, for Started Task Control, as the third position).

Note: A started task can be given a different name, or identifier, when it is started. For example, if SA OS/390's procedure is called `INGEIO` but operators want to use `IOOPS`, they can start SA OS/390 with the following command:

```
START INGEIO.IOOPS,SUB=MSTR
```

From then on, only IOOPS or INGEIO.IOOPS can be used to identify the SA OS/390 task in operator commands, while INGEIO will be unknown.

The ** in the RACF profile lets OS/390 find the correct profile whether or not there is an identifier.

Then we supply the default access level, called UACC (for Universal Access). (The UACC levels are NONE, READ, UPDATE, CONTROL, and ALTER). UACC(NONE) keeps every user away from this command, unless we specifically give them authority.

```
SETROPTS RACLIST(OPERCMDS) REFRESH
```

SETROPTS

This command updates the RACF profile information that the system is using at this moment. This updates all of the access control information relating to operator command protection, since we've just been changing it. After this command, or after the next IPL, no unauthorized user will be allowed to use any SA OS/390 command from any console interface.

Granular Control: You might want to allow some operators to have complete control over SA OS/390, but keep others from starting, stopping or cancelling it. These instructions allow you to pick and choose exactly which operators are allowed to issue which console commands.

The following commands will prevent any console user from controlling I/O operations:

```
RDEFINE OPERCMDS MVS.START.*.ioproc.**    UACC(NONE)
RDEFINE OPERCMDS MVS.STOP.*.ioproc.**     UACC(NONE)
RDEFINE OPERCMDS MVS.MODIFY.*.ioproc.**   UACC(NONE)
RDEFINE OPERCMDS MVS.CANCEL.*.ioproc.**   UACC(NONE)
RDEFINE OPERCMDS MVS.FORCEARM.*.ioproc.** UACC(NONE)
RDEFINE OPERCMDS MVS.FORCE.*.ioproc.**    UACC(NONE)
```

Notes:

1. See "Mixing Generic and Specific Resource Names" on page 149 for a description and examples of when to use specific resource names.
2. OS/390 does not have the ability to distinguish between different product commands entered with the MODIFY operator command. So an operator who has been authorized to enter a non-intrusive command such as MODIFY IOOPS,DISPLAY TIMEOUT is also authorized to enter a more impacting command, such as MODIFY IOOPS,BLOCK (C0) * FORCE.

   Much greater granularity is available to protect the use of SA OS/390 commands accessed through the application programming interface (API), which include access from ISPF, NetView, and the graphical workstation. If you need more levels of control, do not permit access of SA OS/390 commands through the console interface. Use API controls instead.
3. These examples continue using the practice of replacing STC and JOB resource names with "*". You do not need to use the resource names with JOB, but you might still want to use "*" instead of STC.

Define Consoles to RACF: In the CONSOLxx member of SYS1.PARMLIB, there are several statements and attributes that you should be aware of:

The DEFAULT statement

can be used to control the initial security mechanism for consoles. The options are LOGON(AUTO), LOGON(OPTIONAL), and LOGON(REQUIRED).

The default value is OPTIONAL, which is the least intrusive to most installations' current operating procedures. The most secure is REQUIRED, and the easiest is AUTO.

The LOGON attribute is used to describe how and when operators must use the LOGON operator command. If the console operator does not use LOGON, the console has the authority level that is specified for it in the AUTH attribute of the CONSOLE statement.

The NAME attribute of the CONSOLE statement

is used to give each console a unique name.

You use the console name to tell RACF about the console. Rather than manage a new type of profile, RACF considers a console a user. So you tell RACF about a console by adding a RACF user profile with the console name as the user ID, like this:

```
ADDUSER (consoleid) PASSWORD(password)
```

Like other passwords with RACF, the password is initialized expired, so each console and console user is prompted to change the password the first time they logon.

The AUTH attribute of the CONSOLE statement

describes the *old* authorization level. If you don't use OPERCMD authorization, you are using AUTH levels. The levels are INFO, SYS, IO, CONS, and MASTER. The ALL authority is also available, which includes all of the SYS, IO, and CONS authorities.

The operator commands in question (such as START, STOP, MODIFY) are all in the SYS class of commands.

Typically, most consoles are given ALL or MASTER authorities, which means very little console security is in place. Using LOGON(AUTO) or INFO authority, or both, might be a big change to some installations, but it will provide much greater security for console operations.

Operators can use the LOGON command even when a console is already logged-on to another *userid*, although the previous *userid* will be automatically logged-off. In this way, operators can walk up to any console, issue LOGON with their *userid* and password, issue their desired operator commands, then LOGOFF before leaving. They can LOGON even while logged on to another console. This is so that operators are not locked out if they haven't issued LOGOFF from their usual console, but you could also have several consoles used with the same *userid* to share the same authorization class. It might not be as convenient as many operators are used to, but it is secure.

Define TSO Users Who Use the CONSOLE Command to RACF: You have to create a RACF profile for each TSO user who intends to use the TSO CONSOLE command.

```
RDEFINE OPERCMDS MVS.MCSOPER.userid UACC(NONE)
PERMIT MVS.MCSOPER.userid CLASS(OPERCMDS) ID(userid) ACC(READ)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

These commands tell RACF about a profile, in the class of operator commands, that control access for the user *userid* to the Master Console Services for operator commands (MCSOPER). Initially, no one has access to the ability, but the PERMIT command specifies that user *userid* has access to the profile, thus the ability is permitted.

Like always, it doesn't take effect until after the `SETROPTS RACLIST(OPERCMDS) REFRESH` command.

Note: Some installations need to give RACF more information than normal for users who intend to use the CONSOLE command. Often this requires additional TSO parameters, such as accounting information, region size, or logon procedure name.

Allow Consoles or TSO Users to access SA OS/390 Console Commands: Once the resource and console profiles are defined, controlling access to the commands is simple:

```
PERMIT MVS.*.*.ioproc.** ID(userid) CLASS(OPERCMDS) ACC(UPDATE)
```

This command permits user *userid* to access the function described by the profile `MVS.*.*.ioproc.**`. The user is given `UPDATE` authority. This authority is sufficient for the START, STOP, MODIFY, and CANCEL commands. The FORCEARM and FORCE commands require `CONTROL` authority.

To allow user *userid* to issue MODIFY commands to I/O operations, but not START or STOP it, use:

```
PERMIT MVS.MODIFY.*.ioproc.** ID(userid) CLASS(OPERCMDS) ACC(UPDATE)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

Always make certain that you update the in-storage security information with the SETROPTS command after you have changed resource or profile information.

Mixing Generic and Specific Resource Names: You do not have to control each command with a specific resource name. RACF looks for a specific resource name before looking for a generic resource name. If a specific resource name is found, authorization is checked for it, otherwise authorization is checked against the generic name. You could easily protect all operator commands, then further protect the MODIFY command, then further protect modifying I/O operations with:

```
RDEFINE OPERCMDS MVS.** UACC(NONE) RDEFINE OPERCMDS MVS.MODIFY.**
UACC(NONE) RDEFINE OPERCMDS MVS.MODIFY.*.ioproc.** UACC(NONE)
```

When describing granular control in "Granular Control" on page 147, we defined specific resource names (with a "*" for `JOB`or `STC`) for each operator command that can control SA OS/390, such as START or MODIFY.

One approach is to offer two types of SA OS/390 access: one for operators who can issue the MODIFY command, but no other commands and one for other operators who have access to all commands. In such a case, you might make a generic access resource and a specific resource for MODIFY, and grant MODIFY access to the first group and both MODIFY and generic to the second group, like this:

```
RDEF OPERCMDS MVS.*.*.ioproc.**      UACC(NONE)
RDEF OPERCMDS MVS.MODIFY.*.ioproc.** UACC(NONE)
PE MVS.*.*.ioproc.**      ID(userid) CLASS(OPERCMDS) ACC(CONTROL)
PE MVS.MODIFY.*.ioproc.** ID(userid) CLASS(OPERCMDS) ACC(UPDATE)
PE MVS.MODIFY.*.ioproc.** ID(modify_id) CLASS(OPERCMDS) ACC(UPDATE)
```

If most operators need a "base level" of access and only a few need to issue the other commands, and you want to easily add and remove these base level commands, you can define specific resource names for the commands that need special protection, and the generic resource is used for access to the base. This example defines a general authorized operator *userid* to START, STOP, and CANCEL, and *modify_id* to have access to all commands:

```
RDEF OPERCMDS MVS.*.*.ioproc.**       UACC(NONE)
RDEF OPERCMDS MVS.MODIFY.*.ioproc.**   UACC(NONE)
RDEF OPERCMDS MVS.FORCEARM.*.ioproc.** UACC(NONE)
RDEF OPERCMDS MVS.FORCE.*.ioproc.**    UACC(NONE)
PE MVS.MODIFY.*.ioproc.**   ID(modify_id) CLASS(OPERCMDS) ACC(UPDATE)
PE MVS.FORCEARM.*.ioproc.** ID(modify_id) CLASS(OPERCMDS) ACC(CONTROL)
PE MVS.FORCE.*.ioproc.**    ID(modify_id) CLASS(OPERCMDS) ACC(CONTROL)
PE MVS.*.*.ioproc.**        ID(modify_id) CLASS(OPERCMDS) ACC(UPDATE)
PE MVS.*.*.ioproc.**        ID(userid) CLASS(OPERCMDS) ACC(UPDATE)
```

This method allows you access from a console to all of I/O operations or to none of it.

RACF-Based NetView Environments

The following information does not address general administration of RACF-based security in the NetView environment or all that is involved in deciding to migrate to RACF-only security. To learn about that, see *Resource Access Control Facility (RACF) Security Administrator's Guide*

Full RACF checking for SA OS/390 includes:

- Verifying that an operator (whose identity was already authenticated at logon time) is permitted to perform a specific function with a specific operand or option
- Using NetView to check that the operator who originally issued the command is authorized. This is more secure than checking to see that the operator who sends the command is authorized on the system where the command is sent.

To enable full RACF authorization checking, set these option statements with the following NetView keywords and values:

Name of NetView Member	Function within NetView	Keyword and Value	Purpose of Change
DSIDMN	NetView domain definition member	OPERSEC=SAFDEF	Set NetView to use RACF for general operator authentication and authorization. The broadest of all security settings.
		OPSPAN=SAF	Sets authorization of the START SPAN command. This value is the default and is the only possible choice when OPERSEC= SAFDEF is set.
		CMDAUTH=SAF	Sets NetView to use RACF for general command authorization.
		BACKTBL=*table_name*	Provides a backup authorization mechanism to be in place when RACF is not available.

Name of NetView Member	Function within NetView	Keyword and Value	Purpose of Change
		AUTHCHK=SOURCEID	Sets NetView to perform authorization against the ID of the original command issuer.
		SAFNODEC={PASS|FAIL}	Tells NetView what decision it should make on its own when RACF is not available or under certain other circumstances. See the NetView documentation for the consequences of using the default PASS or changing it to FAIL.
DSIUINIT	Initialization member for the DSIUDST task that handles RMTCMD data services	RMTSECUR...SAF	Causes RACF authorization checking for the initial RMTCMD request and ENDTASK requests, based on the RACF RMTOPS class
DSICMD	NetView command definition sample	CMDCML...SEC=DE	Controls authorization at an individual command level.

NetView operators' authorizations are set up by:

- Creating a profile for each instance of NetView in the APPL class. This is done with RDEF and by using as the resource name the DOMAINID from the NCCFID statement in DSIDMN.
- Issuing RACF PERMIT commands for each operator, specifying the resource name (as defined with RDEF), the class (APPL), the operator ID, and the access level for the operator.

Authorizations are stored in a NetView segment for each operator. This is done by specifying a value for CTL with the RACF ALTUSER command with OPERSEC=SAFDEF. Assuming that operator *AUTBASE* has been defined by

```
AUTBASE    OPERATOR   PASSWORD=XYZ123
PROFILEN   AOFPRFAO
```

in the definitions pulled into DSIOPF and

```
AOFPRFAO   PROFILE    IC=AOFPARCR
AUTH       CTL=GLOBAL
OPCLASS    1,2
END
```

in the operator profile, then the new definitions for the RACF environment are:

```
ADDUSER AUTBASE PASSWORD(XYZ123)
ALTUSER AUTBASE NETVIEW(IC(AOFPARCR) CTL(GLOBAL) OPCLASS(1,2))
```

To authorize a NetView operator to start a span (assuming that OPSPAN=SAF is set), you need to:

- Create a profile for the span in the NETSPAN class with RDEF
  - Other existing definition steps for the span such as listing the span name in the SPANLIST statement in DSISPN are unchanged.

- Set up for each operator PERMITS with access authority appropriate for the function that the operator is allowed to execute within that span.
  - UPDATE authority is often the one required to make changes in important commands like VTAM VARY.
- Authorize use of the individual commands.

To check an operator's access to a certain resource, you can use either:

- NetView QRS command or DSIQRS macro, in which the check can be by resource name or by RODM object ID
- CNMSCOP service, which determines whether a particular user is authorized to use a specific command, keyword, or value combination. This is available from a high-level language.

To assign a default extended MCS console name for operators, you can use either:

- The CONSNAME field of the NetView segment when OPERSEC=SAFDEF and
  ```
  ALTUSER AUTBASE NETVIEW(...CONSNAME(AMZ$123)...)
  ```
- GETCONID CONSOLE=x where the console ID is x. This method is preferred to the default name in CONSNAME and must be used when the default name is already in use or defaulting to the NetView task name or operator ID (if nothing is specified in CONSNAME).

Protecting Commands Defined to NetView

Some SA OS/390 commands can be defined to NetView. After these have been activated with SETR CLASSACT(NETCMDS) and SETR GENERIC(NETCMDS), you can establish their security restrictions by issuing RACF RDEF definitions for each command, keyword, and value and combination of these.

When this is used, the resource name is in the form

```
NETID.LUNAME.COMMAND.KEYWORD.VALUE
```

where

NETID
The netID from the last VTAM activation. You may chose to use something generic here if your situation makes that preferable.

LUNAME
Name specified in the NCCFID statement in DSIDMN, the NetView domain ID

COMMAND
Name on the CMDMDL statement in DSICMD for the command. This is not a synonym defined by the CMDSYN statement

KEYWORD
Name required only if you desire to have operand and value-level checking

VALUE
Name required only if you desire to have operand and value-level checking

Then issue RACF PERMIT commands for each operator, specifying the resource name (as defined with RDEF), the class (APPL), the operator ID, and the access level for the operator. The standard access level required for NetView commands is READ.

In the NetView CMDMDL statement in the DSICMD member, the keyword SEC can have one of these values:

BY Indicates that NetView should bypass authority

CH Indicates that NetView should check authority

DE Indicates that NetView should defer a decision about checking until the command is issued and make the decision then based on whether the command is or is not issued from the automation table. If the command is issued from the automation table, authorization is checked if AUTOSEC=CHECK is set by use of the DEFAULTS command and not checked if AUTOSEC=BYPASS. If the command is not issued from the automation table, authorization is always checked.

When the SEC keyword is not specified in the CMDMDL statement or there is no such statement for the command or command procedure. SEC=DE is the default.

Granting NetView and the STC-User Access to Data Sets

This section describes what levels of access authorities you need to assign to NetView and to specific started tasks.

Access to XCF Utilities

The CDS recovery as well as some operator commands use the XCF utilities to retrieve couple data set information. Because the DD name SYSPRINT is required by the utilities, but can also be assigned by NetView for holding log data, the call of the utilities is implemented as a started task in the PROCLIB. The input and output data sets used by the started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID (IBM default: STCUSER) to the started task. This user ID must have RACF UPDATE access to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```

where:

hlq is the high-level qualifier for temporary data set defined during the customization

domain is the domain ID of the current NetView

X is I, O, or P

Access to HOM Interface

Sometimes after an IPL an operating system does not know its sender paths to the coupling facilities in the sysplex. In this case the automation functions call the HCD HOM interface to determine the missing path information. As the HOM interface must not run authorized the interface is called via a started task. The input and output data sets used by the started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID (IBM default: STCUSER) to the started task. This user ID must have RACF UPDATE access to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```

where:

hlq
is the high-level qualifier for temporary data set defined during the customization

domain
is the domain ID of the current NetView

X
O or P

Access to IPL Information

The new automation function collecting, displaying, comparing, and deleting IPL information uses two started tasks. It is recommended to run the first started task immediately after an IPL as part of COMMNDxx list processing, to collect the IPL information in the SA OS/390 VSAM data set ″IPLDATA″. The remaining functions are handled by a NetView command. Since the started task as well as the command can delete IPL information both need RACF CONTROL access to the VSAM data set. The started task collecting the information needs RACF READ access to all parmlib members.

When a comparison of IPL information is requested the NetView command schedules the second started task to call ISRSUPC — the compare utility provided by ISPF — as this utility requires fixed ddname. The input and output data sets used by the second started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID (IBM default: STCUSER) to the started task. This user ID must have RACF UPDATE access to the data sets. The data set names are created as follows:

```
hlq.domain.opid.INGPIPLx
```

where:

hlq
is the high-level qualifier for temporary data set defined during the customization

domain
is the domain ID of the current NetView

opid
is the NetView operator ID

x
L, N, or O

Access to Spare Couple Data Sets

Because the CDS recovery allocates and deletes spare couple data sets via an XCF utility the user ID assigned to the started task address space must also have RACF ALTER access to these couple data sets. The names of the spare couple data set are built as follows:

```
hlq.cdstype.CDSnn
```

where:

hlq
is the high-level qualifier for couple data sets defined during the customization

cdstype is ARM, CFRM, LOGR, SFM, SYSPLEX

nn is the sequence number corresponding to the volume entry in the list of volumes

Access to User-Defined Couple Data Sets

In addition, the user ID of the started task address space needs RACF READ access to all user-defined couple data sets. And, when LOGGER recovery is enabled, the user ID needs RACF ALTER access to the LOGR couple data sets as well.

Access to Spare Local Page Data Sets

The new auxiliary shortage recovery allocates and formats spare page data sets. For this reason NetView requires RACF ALTER access to these page data sets. The names of the spare page data set are built as follows:

```
hlq.sysname.Vvolume.Snn
```

where:

hlq is the high-level qualifier for page data sets defined during the customization

sysname is the name of system for which the data set is allocated

volume is the serial number of the volume on which the data set is allocated

nn is a unique sequence number

Restricting Access to INGPLEX and INGCF Functions

This section describes how you can grant and control access of users to the INGCF and INGPLEX commands.

Access to sensitive functions of the INGPLEX and the INGCF commands should be granted to certain operators only. To do this, add one or more value classes to the operator classes of the operators to authorize them to one or all of the functions.

The following key classes and value classes are applicable:

KEYCLASS=INGPLEX VALCLASS=CDS allows for:
- Allocating an alternate CDS via the INGPLEX CDS command
- Controlling the SDUMP options and the SLIP traps sysplex-wide

KEYCLASS=INGCF VALCLASS=STR allows for:
- Forcing the deallocation of a CF structure via the INGCF STRUCTURE command
- Rebuilding a CF structure on another CF via the INGCF STRUCTURE command
- Controlling the SDUMP options and the SLIP traps sysplex-wide

KEYCLASS=INGCF VALCLASS=CF allows for:
- Preparing a CF for removal from the sysplex via the INGCF DRAIN command
- (Re)integrating a CF into a sysplex via the INGCF ENABLE command
- Including KEYCLASS=INGCF VALCLASS=STR

KEYCLASS=INGPLEX VALCLASS=HW allows for:

- – Deactivating the LPAR of a CF via the INGCF DRAIN command
- – Activating the LPAR of a CF (=starting the Coupling Facility Control Code) via the INGCF ENABLE command
- – Including KEYCLASS=INGCF VALCLASS=CF

To activate the authorization check, add the definition of the SA OS/390 Clist INGRCCHK, the key classes and the value classes in the NetView DSIPARM member DSICMD, as in the following example:

```
INGRCCHK CMDMDL MOD=DSICCP,ECHO=N,TYPE=R
     INGPLEX KEYCLASS 3,5
     CDS     VALCLASS 3
     HW      VALCLASS 5
     INGCF   KEYCLASS 3,4
     STR     VALCLASS 3
     CF      VALCLASS 4
```

With these definitions operators with opclass=3 specified in their operator profiles are authorized to issue all functions of the INGPLEX CDS and the INGPLEX CF commands.

Operators with opclass=4 specified in their operator profiles are authorized to issue all functions of the INGCF CF and the INGCF STRUCTURE commands. Value class 4 includes value class 3 of INGCF but not the value class of INGPLEX.

Controlling Access to the Processor Hardware Functions

For processor operations SNMP processor connections and for the Parallel Sysplex enhancements functions that use the BCP Internal Interface, a SAF product such as RACF must be used to define the required resources and grant access to these resources for the authorized NetView users and autotasks.

Allowing NetView to use the BCP Internal Interface

Before you can use the enhanced sysplex functions of SA OS/390 for CF or XCF automation, the hardware resource (HSAET32) must be defined in NetView.

1. Define resource HSA.ET32OAN.HSAET32 in the CLASS FACILITY
2. Permit NetView READ ACCESS to this facility class resource

The following example shows the RACF commands used to define the resource and to grant the required READ access for the NetView user.

```
SETROPTS CLASSACT(FACILITY)
RDEFINE FACILITY HSA.ET32OAN.HSAET32 UACC(NONE)
PERMIT HSA.ET32OAN.HSAET32 CLASS(FACILITY) ID(stcuser) ACC(READ)
```

With the SETROPTS command, the RACF class FACILITY is made available. The second command RDEFINE fully qualifies the HSAET32 resource and sets universal access to none. With the PERMIT command, the RACF defined user *stcuser* gets READ access to this resource. User ID *stcuser* must be the user ID associated with your NetView started task. If you start NetView as a regular job, the user ID submitting the job must be authorized for the resource.

Note, that you can use a wildcard character to specify the resource more generic if that is suitable for your environment.

Access to the CPCs

Each processor (CPC) defined in your SA OS/390 policy data base must have a corresponding resource profile defined with your SAF product. Note, that this does not apply for processors defined with a connection type SNMP and INTERNAL.

The skeleton of the CPC resource is:

```
HSA.ET32TGT.netid.nau
HSA.ET32TGT.netid.nau.lpar
```

The netid.nau part of the resource name corresponds with the netid.nau definition of the CPC entry specified in the customization dialog. The period between netid and nau is part of the resource name. For LPAR protection define a resource with the netid.nau.lpar specification.

The following example shows how to define a CPC resource in RACF.

```
RDEFINE FACILITY HSA.ET32TGT.DEIBMD1.X7F1F30A UACC(NONE)
```

The CPC with netid DEIBMD1 and nau X7F1F30A is defined as a resource in the RACF class facility with a universal access attribute of NONE.

Note, that you can use a wildcard character to specify the resource more generic if that is suitable for your environment.

Levels of CPC Access

The following lists the access levels and their meaning for the CPC resources.

- READ — Retrieve, get configuration information from the CPC
- WRITE — Update, set configuration information of the CPC
- CONTROL — Issue operations management commands of the CPC

Note: this access level scheme is for the CPC and its LPARs.

Defining the CPC access lists

With SA OS/390, several NetView autotasks need to be authorized to access the CPCs that are defined in the customization dialog.

The following NetView autotasks need to be authorized with access level CONTROL for **all** defined CPCs and all its LPARs:

- The XCF autotasks
- The autotasks defined with SYN %AOFOPXCFOPER% in automation table member AOFMSGSY
- The HW interface autotasks AUTHWxxx

The AUTXCFxx autotasks plus the additional ones from %AOFOPXCFOPER% are used internally once INGCF drain or INGCF enable is invoked by an authorized user. The IXC102A message automation is also performed by these autotasks.

The autotasks used for the HW interface initialization and communication also need to be authorized. Use access level CONTROL for the AUTHWxxx autotasks in your environment.

The following example shows how to permit access to a CPC resource in RACF:

```
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

Operator Profiles

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A.

LPAR access example:

```
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A.* CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A and all its defined logical partitions.

Implementing Granular Hardware Access

By giving operators READ access to a CPC resource and CONTROL access only to LPARS according to the business needs, a flexible security scheme can be implemented.

Defining the UNIX Segments (OMVS) for UNIX System Services Automation

Depending on the operator security definition of NetView, one or more UNIX segments must be defined. These OMVS segments can have a root UID (0) or a non-root UID. To run non-root means more setup.

When using OPERSEC=MINIMAL, NETVPW, or SAFPW, one OMVS segment must be defined. This is the segment for the user ID running NetView.

When using OPERSEC=SAFCHECK, or SAFDEF (user level security), the following operator IDs need a UNIX segment:

- AUTWRK01-NN
- RPCOPER
- MONOPER
- AUTRPC
- AUTO1
- AUTSYS (backup task for AUTRPC and AUTO1)
- AUTBASE (backup task for AUTRPC and AUTO1)
- all tasks receiving actions from the MAT for UNIX resources, usually these are the work operators

Using the OMVS Segment with Root UID

This is the easiest way for setting up the OS/390 UNIX segment. To give it a UID of 0 (root user), enables this user to operate without restrictions. This segment must also be permitted to the RACF facility class BPX.DAEMON (if defined).

Note: Each user who can change NetView CGlobals may be able to issue UNIX System Services commands under a root user ID.

Using the OMVS Segment with Non-Root UID

If you want to reduce the number of UID 0 users, it is possible to define a setup without UID 0 with some restrictions.

If you are using a setup with non-root UID, the OMVS segment must be defined in the following way:

Monitoring

- For process monitoring:

  Define read access to `SUPERUSER.PROCESS.GETPSENT`

  This allows a user ID to see all processes. If the user ID performing the monitoring is not allowed to check all processes, the automation may assume that the start was not successful and restarts the application. This will result in many instances.
- For file or filesystem monitoring:

  Define read access to `SUPERUSER.FILESYS`

  This allows a user ID to get access to all files in the UNIX file system. If the user ID performing the monitoring is not allowed to check all files, the automation may assume that the resource is unavailable.
- Give access to any resource that user-written monitoring routines may use
- For user-defined monitoring, see “Command Execution (INGUSS)”. (User defined monitoring is performed with the command INGUSS.)

Command Execution (INGUSS)

- Give the OMVS segment the ability to switch to any user ID associated with OS/390 UNIX resources (access to `BPX.SRV.userid` or `BPX.SUPERUSER` to start root programs).
- Depending on your security environment the OMVS segment may need access to `BPX.DAEMON`.
- The OMVS segment must be authorized to perform all the commands that are specified in the customization dialogs. For an overview of authorizations for non-root users, refer to the chapter that explains UNIXPRIV class profiles in *z/OS UNIX System Services Planning*.

Restrictions for Non-Root UID Setup

There is an MVS identity and an OS/390 UNIX identity. Without a UID 0 you cannot switch the MVS identity. If a user needs access to certain MVS data sets, you may not start the application with INGUSS. You may get in trouble when automating OS/390 UNIX resources that require a UID of 0 (for example, the inetd). The OMVS segments without UID 0 are normally not able to switch to a root user in order to perform actions. SA OS/390 standard monitoring will work. For example, if you allow the OMVS segment to switch to UID 0 (by defining read access to BPX.SUPERUSER), you could also assign it a UID of 0.

Creating an OMVS Segment by Submitting a Job

Creating OMVS segments can be done by submitting a job, as shown in Figure 31 on page 160.

The NOPASSWORD option prevents unauthorized logins.

This OMVS segment must be authorized to set the job name (read access to `BPX.JOBNAME`). Otherwise, the started address spaces have the same jobname as NetView. When the job name can be set, the newly created address space has the job name INGCUNIX.

```
//*
//ADDUSER  EXEC PGM=IKJEFT01
//*
//SYSTSPRT DD SYSOUT=*
//SYSLBC   DD  DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD *
  ADDUSER STCUSER +
          NOPASSWORD+
          UACC(NONE) DFLTGRP(AUTGRP) +
          OMVS(UID(0000000) HOME('/')  PROGRAM('/bin/sh')) +
//*
//COUSERS  EXEC PGM=IKJEFT01
//*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  CO      STCUSER GROUP(USERS) AUTH(USE)
//*
```

Figure 31. Job Example of Creating an OMVS Segment

Installing the SA OS/390 Password Protection Feature (Optional)

The SA OS/390 password protection feature allows SA OS/390 to use a Security Authorization Facility (SAF) program such as RACF to manage gateway operator IDs and passwords. This feature is intended for systems that establish gateway sessions with two or more other systems. After installing SA OS/390 and the SAF, the steps for installing the SA OS/390 password protection feature are as follows:

1. Edit the DSIDMN member of the NetView DSIPARM data set.
2. Define gateway operator IDs to the SAF program.
3. Define entries for the SA OS/390 password data set.
4. Customize SA OS/390 password protection for a shared SAF environment.
5. Set up I/O operations security
6. Recycle NetView

The steps are described in more detail in this section, and reference the example automation network shown in Figure 32, which shows a focal point system connected to two target systems. All systems are named by their NetView domain IDs: ING01, ING02, and ING03.

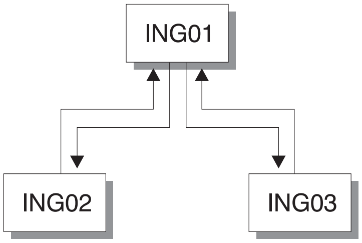


Figure 32. Automation Network Example for SA OS/390 Password Protection

Step 1 — Edit the DSIDMN Member of the NetView DSIPARM Data Set

Change the VERIFY parameter of the OPTIONS statement to VERIFY=MAXIMUM. See *Tivoli NetView for OS/390 Security Reference* for information about the OPTIONS statement.

Step 2 — Define Gateway Operator IDs to the SAF Program

A Security Authorization Facility (SAF) program such as RACF manages all user IDs that are associated with a system or group of systems. Each system in an automation network can have its own password data set; or several systems in the network can share one password data set.

A security administrator usually manages the SAF for a DP enterprise, creating user IDs and assigning passwords. To define gateway operator IDs and passwords for your automation network, do the following:

1. Identify the gateway operator IDs that each system in your automation network needs. For example, referring to Figure 32 on page 160, systems ING02 and ING03 would both need a gateway operator ID (GATING01) to enable the logon from the outbound gateway operator on system ING01. System ING01 would need two inbound gateway operator IDs: one to enable the logon from system ING02, another to enable the logon from system ING03.
2. Ask your security administrator to define these gateway operator IDs to the SAF program on the appropriate systems. Also, request the passwords for those gateway operator IDs.

If you use RACF, and your DP enterprise does not have a security administrator, see *Resource Access Control Facility (RACF) Command Language Reference* for information about the ADDGROUP and ADDUSER commands.

Step 3 — Define Entries for the SA OS/390 Password Data Set

The SA OS/390 password data set contains only gateway operator IDs and passwords for systems to which your system connects. To create the SA OS/390 password data set, ING.ING01.PASSWORD, and define entries for it, do the following:

1. Allocate VSAM space for the SA OS/390 password data set.

   Copy and edit job INGESYSA, which resides in the SINGSAMP data set, to allocate space for the SA OS/390 password data set.

   Before you run job INGESYSA, do the following:

   a. Enter your JOB statement information by modifying the example job card.
   b. Enter the correct data set names and any other information that is unique to your environment.
   c. Comment job step 1 unless you want to delete the existing SA OS/390 password data set. If you want to delete the existing SA OS/390 password data set, leave job step 1 without comment signs.

      Note: Job step 1 calls member INGESYSA, which also deletes the couple data set. If you want to delete only the SA OS/390 password data set, edit the INGESYSA member, and be sure that the DELETE statement for the SA OS/390 password data set (ING.ING01.PASSWORD) is not a comment, and the DELETE statement for the coupling facility data set is commented.
   d. Comment job step 2.
   e. Remove comment signs from job step 3. Job step 3 uses member INGESYSA to allocate space for the SA OS/390 password data set, ING.ING01.PASSWORD.
2. Define the SA OS/390 password data set to NetView by adding the following statement to the NetView application startup procedure:

```
//AOFPSWD   DD   DSN=ING.ING01.PASSWORD,
//             DISP=SHR,AMP=AMORG
```

ING.ING01.PASSWORD is the default SA OS/390 password data set for which space was allocated earlier.

3. Add gateway operator IDs and passwords to the SA OS/390 password data sets.

   Use the SA OS/390 INGPW command on each system in the automation network to add gateway operator ID and password entries to their SA OS/390 password data sets. For example, a system programmer at system ING01 would enter the following commands to add gateway operator IDs and passwords for systems ING02 and ING03:

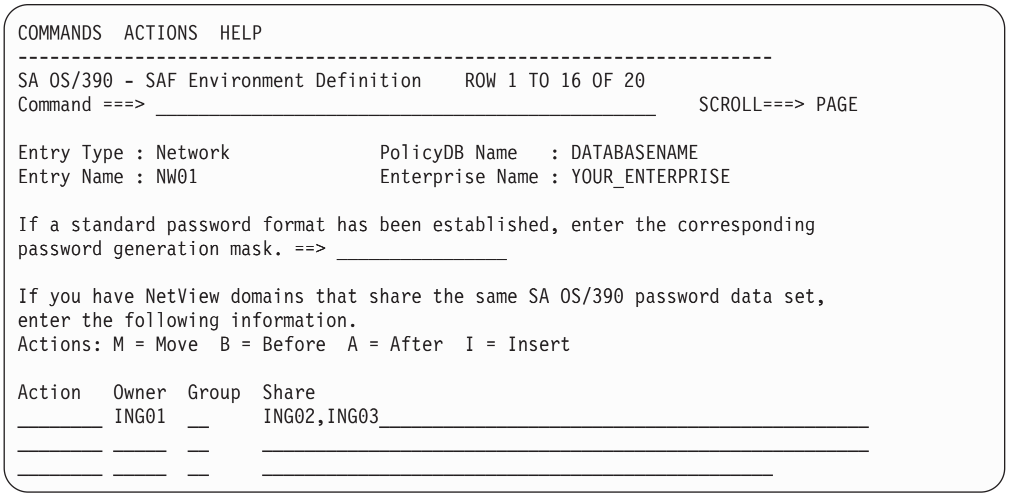
   ```
   INGPW GATING01 ING02,INIT=ABCPASS1
   INGPW GATING01 ING03,INIT=ABCPASS2
   ```

   GATING01 is the ID for the outbound gateway operator on system ING01. ABCPASS1 and ABCPASS2 are the passwords the system programmer received from the security administrator.

   See *System Automation for OS/390 Programmer's Reference* for detailed information about the INGPW command and an example of how to define entries for a shared SAF environment.

Step 4 — Customize SA OS/390 Password Protection for a Shared SAF Environment

If your automation network shares a RACF or other SAF data set, you must define how the data set is to be shared. This applies only to systems that are connected to two or more other systems, which share the same RACF or other SAF data set. Use the customization dialog to specify the systems that share the data set and the system that owns the outbound gateway operator that connects to these systems. For example, if the automation network shown in Figure 32 on page 160 were sharing a RACF or other SAF password data set, a system programmer at system ING01 would use the customization dialog to make the following entries in the System Authorization Facility (SAF) Environment Definition panel as shown in Figure 33.



```
COMMANDS  ACTIONS  HELP
----------------------------------------------------------------------
SA OS/390 - SAF Environment Definition    ROW 1 TO 16 OF 20
Command ===> ____________________________________________    SCROLL===> PAGE

Entry Type : Network              PolicyDB Name   : DATABASENAME
Entry Name : NW01                 Enterprise Name : YOUR_ENTERPRISE

If a standard password format has been established, enter the corresponding
password generation mask. ==> ________________

If you have NetView domains that share the same SA OS/390 password data set,
enter the following information.
Actions: M = Move  B = Before  A = After  I = Insert

Action   Owner  Group  Share
________ ING01  __     ING02,ING03_________________________________________
________ _____  __     ____________________________________________________
________ _____  __     ________________________________________________
```

Figure 33. Customizing Password Protection: SAF Environment Definition

The Action field is for Actions that change the order in which the rows of information are presented. Valid actions are M (Move), B (Before), A (After), and (I) Insert. The Owner field shows the system that owns the outbound gateway operator that connects to systems ING02 and ING03. The Share fields show the systems to which the ING01 system connects. The Group fields show the group name assigned to multiple systems.

The password generation mask field describes the format to be used when generating new passwords for gateway autotasks. See *System Automation for OS/390 Defining Automation Policy* for more information on the INGPW command, and on defining a password mask and for information about customizing policy for the SA OS/390 password protection feature.

Step 5 — Recycle NetView

Stop NetView and restart it. See *Tivoli NetView for OS/390 User's Guide* for information about stopping and starting NetView.

Step 6 — Define a RACF Profile for I/O Operations

Assign authorization levels using RACF/SAF for individual commands or generically for all commands. Use the RACF RDEF command with a class of FACILITY.

Function	Command
To define the profile for the PROHIBIT command	RDEF FACILITY IHV.PROHIBIT
To define a profile that would allow all users to enter a command (for example, UNLOCK)	RDEF FACILITY IHV.UNLOCK UACC(READ)
To permit the use of generics for a Class of Service facility	SETROPTS GENERIC FACILITY
To prevent unauthorized use of commands you can enter this RACF command to prohibit use of commands	RDEF FACILITY IHV.* UACC(NONE)

Note: If you have prohibited all user IDs from using these commands, you must explicitly assign RACF authorization to designated user IDs.

Assign RACF Authorization

To give RACF authorization to a user ID, enter the RACF PERMIT command and its parameters.

Assign a Profile Parameter: The profile parameter is `IHV`*commandname*, where:

- `IHV.` is the three-character ID, followed by a period (.).
- *commandname* is the name of the command

Notes:

1. The profile parameter (for example, IHV.ALLOW, IHV.VARY, IHV.REMOVE.SWITCH) determines the authorization level of the user ID identified in the ID parameter.
2. The ACCESS parameter identifies the authorization given.

   You can use an asterisk to designate a generic class on the PERMIT parameters. For example, to allow all users to send all commands that require read authority, enter:

   ```
   PERMIT IHV.* ACCESS(READ) CLASS(FACILITY)
   ID(*)
   ```

Assign Authorization by ACCESS Level

You can authorize a user ID to enter one command at a given access level by entering one command.

For example, to allow a user (SUWAJDA) to send commands requiring control authorization, enter:

```
PERMIT IHV.* ACCESS(CONTROL) CLASS(FACILITY)
ID(SUWAJDA)
```

For example, to authorize another user (FISHER) to enter all commands that require the update authorization, enter:

```
PERMIT IHV.* ACCESS(UPDATE) CLASS(FACILITY)
ID(FISHER)
```

Assign Authorization by Command: You can use the PERMIT command to let all users send individual commands. For example, to authorize everyone to use the Unlock command, enter:

```
PERMIT IHV.UNLOCK ACCESS(READ) CLASS(FACILITY)
ID(*)
```

To authorize a user (DONC) to send all connectivity commands with the Noforce option, enter:

```
PERMIT IHV.* ACCESS(UPDATE) CLASS(FACILITY)
ID(DONC)
```

Use Specific Profile Names: Either specific profile names or generic profile names can be used in the PERMIT command. Use specific profile names to authorize use of specific I/O operations commands.

For example, to authorize a user (PHILOP) to use only the Allow and Prohibit commands with the Noforce option, enter:

```
PERMIT ING.ALLOW ACCESS(UPDATE) CLASS(FACILITY) ID(PHILOP)
PERMIT ING.PROHIBIT ACCESS(UPDATE) CLASS(FACILITY) ID(PHILOP)
```

On the NMC focal point the following is necessary to define users and access levels to RODM:

1. Define a general resources class named RODMMGR. This is the default class name used in EKGCUST initialization member for RODM.
2. Define instances of the RODMMGR resource class, for example,

   ```
   RDEF EKGXRODM1 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM2 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM3 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM4 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM5 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM6 CLASS(RODMMGR) UACC(NONE)
   ```

For more information on the RACF commands, see *Resource Access Control Facility (RACF) Command Language Reference*.

Establishing Authorization With Network Security Program (NetSP)

If you have installed NetSP, you can create an authorization system requiring only one sign on for each user. With it, a user who logs on from a workstation has access to RACF-protected host applications. These include 3270 emulation and log on scripts and APPC communications. This authorization is controlled by NetSP's PassTicket, which is recognized by the SAF-based security system and is valid for a fixed period of time.

To establish authorization for your users, you need to create in NetSP recorded input files as log on transfer scripts. This is done either by recording keystrokes in the emulator session or by entering them directly in a file with a text editor. How to do this is described in *Network Security Product Secured Network Gateway Guide*.

Appendix B. Syntax for INGTOPOF File

The INGTOPOF file contains configuration information for the SA OS/390 topology manager. It must reside in any of the data sets allocated under the DSIPARM concatenation. The records of the file consist of a keyword with one or more parameters. Comment lines must start with an asterisk (*). A '+' at the end of a line indicates that the record is continued in the next line.

The following keywords can occur in the INGTOPOF file: SYSPLEX, PROCOPS, LOCATION, ANCHOR, BLDVIEWS, OPTION, and TEMPLATE.

The SYSPLEX Statement

For every sysplex, the SA OS/390 topology manager must be told which systems of the sysplex are able to communicate with it. This is done with the SYSPLEX statement according to the following format:

```
                             ┌─────────────────────────────┐
                             ▼                             │
►►──SYSPLEX──sysplex_name────┬─────────┬──domain_id────────┴──────────────────────────►◄
                             └─net_id.─┘
```

The *sysplex_name* must be different from every name that you specify in a PROCOPS statement (see "The PROCOPS Statement" on page 168). The systems must be identified to the SA OS/390 topology manager by their NetView domain ID. If the *net_id* is omitted, it is assumed to be the same as that of the focal point. The INGTOPOF file must contain at least one SYSPLEX statement; in particular, you cannot have a PROCOPS statement in the INGTOPOF file without a SYSPLEX statement.

The SA OS/390 topology manager tries to contact the systems in the order in which they appear in the list. When it finds a system that contains a functional SA OS/390 topology agent, it searches no further, but gathers the SA OS/390 information from the automation manager through this SA OS/390 topology agent. It then stores the retrieved information within RODM, prefixing all resource names with the *sysplex_name* that it found in the SYSPLEX statement.

It follows from this that the order in which the domains are specified should reflect eventual decisions about primary and backup systems for communication with the SA OS/390 topology manager. Also, the sysplexes as defined in the INGTOPOF file must correspond to the sysplex groups in the policy database.

Since stand-alone systems are treated as sysplexes, they must also be introduced to the SA OS/390 topology manager by a SYSPLEX statement. In this case, the list of domain IDs will comprise just one item.

If you want to have a network anchor for a system, this system's domain ID must be included in the SYSPLEX statement.

The PROCOPS Statement

With this statement, you specify a focal point for processor operations and its backup focal point. It has the following format:

```
▶▶──PROCOPS──procops_name──focal_point──backup_focal_point──────────────────▶◀
```

The *procops_name* must be different from every name that you specify in a SYSPLEX statement. The focal point processor and its backup must be identified to the SA OS/390 topology manager by a NetView domain ID. If the *net_id* is omitted, the SA OS/390 topology manager assumes it to be identical to that of its own focal point.

There must be at least one SYSPLEX statement in the INGTOPOF file if you want to insert a PROCOPS statement.

The LOCATION Statement

The LOCATION statement is used to group system related events, for example, geographically rather than logically. The events that are attached to a LOCATION must be posted to the SA OS/390 topology manager by the user with the INGPOST command. For more information on the INGPOST command, see *System Automation for OS/390 Operator's Commands*.

The Location statement has the following format:

```
▶▶──LOCATION──target_domain──location_name──────────────────────────────────▶◀
```

Examples:

```
*
* TSCF1 thru 3 are in Boeblingen, 4 and 5 are in Perth
*
LOCATION T2 BB_LAB
LOCATION NETOZ.CNMT4 PERTH
LOCATION NETOZ.CNMT5 PERTH
*
* AOCA thru D are in Boeblingen
*
LOCATION AOCPLEX BB_LAB
*
* OZ1 thru OZ4 are in Perth
*
LOCATION OZPLEX PERTH
```

The ANCHOR Statement

The ANCHOR statement serves to define anchors for arbitrary user defined events.

Anchors are objects that occur in RODM, but not within the automation manager. They serve to collect events of a certain type that are to be displayed on the NMC. Anchors play the role of major resources for events of this type, and the events themselves are treated as minor resources of their anchor. The SA OS/390 topology manager automatically creates anchors for certain events such as WTORs. For more information on anchors and events see *System Automation for OS/390 User's Guide*.

With the ANCHOR statement, you can introduce your own anchors for any events. These events must be posted to the SA OS/390 topology manager with the INGPOST command; the anchor must be specified in the command as the major resource (RESOURCE parameter). For more information on the INGPOST command, see *System Automation for OS/390 Operator's Commands*; for information on major and minor resources, see *System Automation for OS/390 Defining Automation Policy*

The ANCHOR statement has the following format:

```
►►──ANCHOR──sysplex_name──anchor_name────────────────────────────────────►◄
```

The BLDVIEWS Statement

A RODM resource can only be displayed on the NMC when it is included in a view. With the BLDVIEWS statement, you can pass data sets (members) that contain view definitions for BLDVIEWS to the SA OS/390 topology manager. The SA OS/390 topology manager will then call the BLDVIEWS tool for (all or some of) these data sets (members) in order to build or rebuild the specified views. The view definitions must be supplied by the customer.

Every BLDVIEWS statement associates one sysplex (as defined by a SYSPLEX statement) or one processor operations focal point configuration (as defined by a PROCOPS statement) with a list of such data sets (members). This enables the SA OS/390 topology manager to rebuild views at runtime only for those sysplexes (sets of target processors) whose SA OS/390 information has in fact changed.

The BLDVIEWS statement has the following format:

```
                             ┌─────────────────────┐
                             ▼                     │
►►──BLDVIEWS──┬─sysplex_name─┬──data_set_or_member─┴──────────────────────►◄
              └─procops_name─┘
```

You can exploit the association of the data sets (members) to sysplexes to reduce the overhead caused by rebuilding views at runtime. Suppose, for example, that all your sysplex views either contain objects from only one sysplex or from all sysplexes. Then you should proceed as follows.

1. For every sysplex, create a separate data set (member) with the view definitions specific for that sysplex.
2. Create one data set (member) for the common views.
3. Code a BLDVIEWS statement for every sysplex, where the list of data sets (members) comprises two items, namely the data set (member) with the views specific for this sysplex, and the data set (member) with the common views.

In this way, the sysplex specific views are rebuilt only when the SA OS/390 resources for the sysplex in question have changed within RODM in such a way that a rebuild is necessary.

For more details on view definitions, see *System Automation for OS/390 User's Guide*.

The OPTION Statement

With the OPTION statements you can:

- control whether or not dependencies and major/minor resource relationships are stored in RODM, and are therefore represented on the NMC, and
- specify the default layout for the automatically generated subviews of group objects.

A separate OPTION statement is required for each option.

The OPTION statement has the following format:

```
          ┌PARENTS───┐
►►──OPTION─┤          ├──────────────────────────────────────────────────►◄
          └NOPARENTS─┘
```

```
          ┌LINKMINOR───┐
►►──OPTION─┤            ├────────────────────────────────────────────────►◄
          └NOLINKMINOR─┘
```

```
          ┌MDL_RADLINK─┐
►►──OPTION─┤            ├────────────────────────────────────────────────►◄
          └MDL_GRID────┘
```

The parameters have the following meaning:

PARENTS
Dependency relationships are stored in RODM (and displayed on the NMC in network views). This is the default.

NOPARENTS
Dependency relationships are not stored in RODM.

LINKMINOR
Relationships between major and minor resources are stored in RODM (and displayed in network views). This is the default.

NOLINKMINOR
Relationships between major and minor resources are not stored in RODM.

MDL_RADLINK
The automatically created subviews are radially arranged. This is the default. The default for this option is defined in RODM.

MDI_GRID
The automatically created subviews are arranged in a grid. The default for this option is defined in RODM.

If you want to use the default values, no explicit OPTION statement is required.

The TEMPLATE Statement

The name displayed beneath a resource on the NMC is the DisplayResourceName field of the resource. It can be customized using the TEMPLATE parameter in the INGTOPOF file. The template entries in the INGTOPOF file control how the DisplayResourceName of a resource is formatted.

When using the locate function on the NMC, it is the DisplayResourceName field of the resource that is compared with the search criteria of the locate for an exact match.

It is not a requirement to have any template parameters in the INGTOPOF file. If no template parameter is found in INGTOPOF, the format of the DisplayResourceName will default to the following:

- `PLEX.SYSTEM.TYPE.SUBSYSTEM EVENT` for major resources
- `PLEX.SYSTEM.TYPE.SUBSYSTEM.MINOR EVENT` for minor resources

To change the format of the default DisplayResourceName, special *type* templates are required to specify how the default DisplayResourceName should be formatted. There are the following two types:

- DRN for major resources
- DRNM for minor resources

Customization of the DisplayResourceName can be defined for all resource types (DRN, DRNM), or individually for each resource type (APL, APLM, APG, APGM).

When a resource is created, the type (for example, APL or APG) of the resource is searched for in the INGTOPOF file to find a matching template.

- If a match is found, the DisplayResourceName will be formatted as specified by the type template in the INGTOPOF file.
- If no match is found, the DisplayResourceName will be formatted using the default.

The major resource types supported by the template parameter in the INGTOPOF file are the following:

APL	applications
APG	application groups
APGP	application groups (sysplex)
SYS	system
SYG	system groups
GRP	groups
ANCH	anchors

As minor resources can be attached to major resources, the following types are also supported by the template parameter in the INGTOPOF file for minor resources:

APLM	application minors
APGM	application group minors
APGPM	application group (sysplex) minors
SYSM	system minors
SYGM	system group minors
GRPM	group minors
HEARTBEATM	heartbeat minors
WTORM	wtor minors

Syntax for INGTOPOF File

TAPEM	tape minors
CFM	coupling facility minors
CDSM	coupled data set minors
ETRM	external timer minors
SYSPLEXM	sysplex minors

All, any, or none of the above type templates can be used.

When user defined anchors are created, the following applies:

- If the format of the default DisplayResourceName is acceptable, no additional template will be required in the INGTOPOF file. The DisplayResourceName is formatted using the default.
- If you customized the format of the DisplayResourceName, it is necessary to create a template for the user-defined anchors, to specify how the DisplayResourceName must be formatted for the user-defined anchors and any minor resources attached to the user-defined anchors.

If the anchor statement `ANCHOR K1 USER` exists in the INGTOPOF file, define the following two type templates in the INGTOPOF file to control the formatting of the DisplayResourceName for the anchor and any attached minor resources:

- USER for the anchor
- USERM for the minor resources attached to the anchor

To define how the DisplayResourceName is formatted, substitution parameters are employed. Substitution parameters can appear in any order. The following substitution parameters are supported:

&STR.	system.type.subsystem
&RES.	subsystem/type/system
&MNR.	minor resource name (minor resources only)
&SUB.	subsystem
&TYP.	type
&SYS.	system
&EVT.	event
&PLX.	sysplex
&DATE.	date
&TIME.	time

If event (&EVT.) is specified as a substitution parameter and no event field exists for the resource, an * is inserted in the DisplayResourceName. If a substitution field does not exist for a resource where a substitution parameter has been specified, then the substitution parameter itself (for example, &SYS. &STR.) will appear in its place in the DisplayResourceName.

Examples

Customizing DisplayResourceName for APLs:

If the requested DisplayResourceName for APLs was system name and subsystem name (for example, SYSX.RODMX), then the following entry would be required in the INGTOPOF file:

```
TEMPLATE APL &SYS..&SUB.
```

Customizing DisplayResourceName for all resources:

If the requested DisplayResourceName for all resources was system name, subsystem name, and event (for example, SYSX.RODMX Event Text), then the following entry would be required in the INGTOPOF file:

```
TEMPLATE DRN &SYS..&SUB. &EVT.
```

Customizing DisplayResourceName for user anchors:

The following anchor statement is found in INGTOPOF file:

```
ANCHOR K1 PLEX1
```

If the requested DisplayResourceName was subsystem, date, and time (for example, RODMX 19 MAY 2002.02:16:45), the following entry would be required in the INGTOPOF file:

```
TEMPLATE PLEX1 &SUB. &DATE..&TIME.
```

The above examples are for major resources. If customization of the DisplayResourceName is also required for minor resources attached to the major resources, then similar template entries in the INGTOPOF file would be required:

- TEMPLATE APLM &SYS..&SUB..&MNR.
- TEMPLATE DRNM &SYS..&SUB..&MNR. &EVT.
- TEMPLATE PLEX1M &SUB..&MNR. &DATE..&TIME.

For an example of the template statements in the INGTOPOF file, refer to “Sample INGTOPOF File” on page 174.

As the DisplayResourceName can now be customized, it is possible to create different resources with the same DisplayResourceName. Although duplicate DisplayResourceNames cause no problems to the NMC or RODM, it will be the responsibility of each installation to ensure that any duplication is correctly processed by any user-written code.

BLDVIEWS creates views containing resources, and can identify resources for inclusion by the MyName field or the DisplayResourceName field of the resource.

- No further change to your BLDVIEWS statements will be required.
- The format of the MyName field may NOT be modified.
- The format of the MyName field is, `PLEX.SUBSYSTEM/TYPE/SYSTEM.MINOR`
- The MyName field may have parts omitted that are not relevant.
- The following are examples of the MyName:

  ```
  PLEX.SUBSYSTEM/TYPE            - major
  PLEX.SUBSYSTEM/TYPE/SYSTEM     - major
  PLEX.SUBSYSTEM/TYPE.MINOR      - minor
  PLEX.SUBSYSTEM/TYPE/SYSTEM.MINOR - minor
  ```

- If you currently use the DisplayResourceName in your BLDVIEWS statements and you are customizing the DisplayResourceName, it will be necessary to review your BLDVIEWS statements to ensure that the correct resources are included in your views.

The RUNOPID statement

When submitting commands via the NMC, the commands are run under the user id of the operator signed on to the NMC at that time.

It is possible to select a predefined user id by using the RUNOPID statement in the INGTOPOF file. When a command is submitted via the NMC for a non-local resource, the command will be run under the predefined user id, and not the user id of the operator signed on to the NMC at that time.

Commands that are issued via the NMC against a local resource are never preceded by a label.

Commands that are issued via the NMC against a non-local resource are preceded by a label. This label has three separate fields:

- Netid
- Domain
- User id

Examples of the label are as follows:

- `Netid:`
- `Netid.Domain:`
- `Netid.Domain/User id:`

To provide an amount of flexibility, the RUNOPID statement has been introduced to the INGTOPOF file. This will allow a predefined user id to be used in the label, rather than the user id of the operator signed on to the NMC at that time.

If the RUNOPID statement exists in the INGTOPOF file, then the associated user id will be substituted in the label.

The syntax of the RUNOPID statement in the INGTOPOF file is

```
RUNOPID user id
```

An example of the RUNOPID statement in the INGTOPOF file is

```
RUNOPID ACDMON
```

If multiple RUNOPID statements appear in the INGTOPOF file, then only the first RUNOPID statement will be used, all subsequent RUNOPID statements will be discarded.

Sample INGTOPOF File

```
*********************************************************************
*
* INGTOPOF sample
*
* The sysplex_name in this example is:  K1
* The sysplex consists of the following four
* domains: IPSNM, IPSNN, IPSNO and IPSNP
*
* The KEY1VIEW and CMNVIEW members contain BLDVIEWS control cards.
* They are necessary for the SA topology manager to create 'views'
* in RODM to display SA resources.
* For more details refer to the SA User's Guide,
* Using the NetView Management Console for SA OS/390,
* Creating Views
```

```
*
* This sample also contains a user defined anchor 'USER' and
* shows the usage of the 'OPTION' and 'TEMPLATE' statements.
*
* For a description of all keywords please refer to the
* System Automation for OS/390 Planning and Installation guide.
*
* Use a trailing '+' for continuation.
*
*********************************************************************
*
SYSPLEX  K1 IPSNM IPSNN +
                  IPSNO +
                  IPSNP
*
BLDVIEWS K1 KEY1VIEW CMNVIEW
*
ANCHOR K1 USER
*
* OPTION NOPARENTS
* OPTION NOLINKMINOR
OPTION MDL_RADLINK
*
*===================================================================*
* To define how the DisplayResourceName is formatted,               *
* substitution parameters are employed. Substitution                *
* parameters may appear in any order. The following                 *
* substitution parameters are supported,                            *
*                                                                   *
* &STR. - SYS.TYPE.SUB                                              *
* &RES. - SUB/TYPE/SYS                                              *
* &MNR. - MINOR RESOURCE NAME (Minor Resources only)                *
* &SUB. - SUBSYSTEM                                                 *
* &TYP. - TYPE                                                      *
* &SYS. - SYSTEM (NULL FOR SYSPLEX RESOURCE)                        *
* &EVT. - EVENT                                                     *
* &PLX. - SYSPLEX                                                   *
* &DTE. - DATE                                                      *
* &TME. - TIME                                                      *
*                                                                   *
* To activate a TEMPLATE statement remove the leading asterisk from *
* the following samples.                                            *
*===================================================================*
*
*TEMPLATE DRN &PLX..&STR. &EVT.
*TEMPLATE DRNM &PLX..&STR..&MNR. &EVT.
*
*TEMPLATE APL &SYS..&SUB.
*TEMPLATE APLM &MNR.
*
*TEMPLATE APG &PLX. &SYS. &RES.
*TEMPLATE APGM &PLX. &SYS. &RES. &MNR.
*
*TEMPLATE APGP &PLX. &RES.
*TEMPLATE APGPM &PLX. &RES. &MNR.
*
*TEMPLATE SYS &PLX..&RES.
*TEMPLATE SYSM &PLX..&RES. &MNR.
*
*TEMPLATE SYG &PLX..&RES.
*TEMPLATE SYGM &PLX..&RES. &MNR.
*
*TEMPLATE GRP &RES. GRP
*TEMPLATE GRPM &RES..&MNR. GRPM
*
*TEMPLATE HEARTBEATM &PLX..&RES. &MNR. &EVT. &DATE..&TIME.
*
```

```
*TEMPLATE WTORM &MNR. &EVT.
*TEMPLATE TAPEM &MNR. &EVT.
*
*TEMPLATE CFM &PLX..&RES. &MNR. &EVT.
*TEMPLATE CDSM &RES. &MNR. &EVT.
*TEMPLATE ETRM &MNR. &EVT.
*TEMPLATE SYSPLEXM &PLX..&RES..&MNR. &EVT. SYSPLEXM
*
*TEMPLATE ANCH &STR. &EVT.
*
*TEMPLATE USER &STR. &DATE. &TIME
*TEMPLATE USERM &MNR. &PLX. &SUB. &DATE. &TIME. &EVT.
*********************************************************************
```

Appendix C. Miscellaneous Information

This section tells you how to do the additional installation tasks involved in using the enterprise monitoring functions of SA OS/390.

Running Two NetViews on the NMC Focal Point System

If your focal point system runs one NetView for automation (Automation NetView) and another NetView for networking (Networking NetView) that includes an NMC focal point system, you must install SA OS/390 on both NetViews. The SA OS/390 installation on the NetView used for networking involves only a subset of SA OS/390 code, called an SA OS/390 satellite, and fewer installation steps are required.

Where the Networking NetView is an enterprise monitoring focal point, the SA OS/390 NetView's DSI6INIT Parm should specify the Networking NetView on the same system as its focal point. The focal point needs to receive heartbeats from the SA OS/390 domain on the same system to set the necessary RODM focal point fields.

Installation of an SA OS/390 satellite is covered as an optional step. See "Step 25: Install an SA OS/390 Satellite" on page 115.

Users and RODM Authorization

When RODM is installed on your system, it is necessary to authorize users and applications to access RODM services. This authorization is accomplished using RACF or an equivalent security application. See *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide* for details about specifying RODM authorization. This section describes any additional user IDs that must be created for system operations enterprise monitoring and indicates whether they require RODM authorization.

Table 21. RODM Authorization for user IDs

User ID	RODM Authorization Required?
NetView Graphic Monitor Facility operators	No
SA OS/390 operators	Yes
User ID for bulk updates from NetView (specified in AOFRODM)	Yes
User ID for GMFHS to connect to RODM (defined when you install GMFHS and RODM)	Yes

Graphic Monitor Facility Host Subsystem (GMFHS) operator IDs are usually created to be the same as NetView operator IDs so that operators can use the same ID and password to log on to GMFHS as they use to log on to NetView. RODM authorization is not required for use of GMFHS, but the IDs may require authorization for other purposes such as using RODMVIEW.

Users and RODM Authorization

Note: If you assign an GMFHS operator ID of OPER1 on the NMC focal point system, GMFHS automatically uses the same GMFHS operator ID on other NetViews within the enterprise as the target for commands.

In addition to logging on to GMFHS, operators using system operations enterprise monitoring need to log on to SA OS/390. You may choose to use the same set of IDs for SA OS/390 as you do for NetView and GMFHS. However, SA OS/390 IDs must be authorized to RODM. Because an ID can only be used to connect to RODM from one application at a time, you should create a unique system operations ID for each operator who connects to RODM from another application.

Verifying Installation of SA OS/390 Satellite (Optional)

You should now test your Networking NetView (with added system operations satellite). An outline procedure for this is:

1. Schedule a testing period. You will require your focal point system and expertise on how the Networking NetView should behave.
2. Shut down your Networking NetView. This means you no longer have any network automation.
3. Start your Networking NetView with the SA OS/390 satellite.
4. Check that it initializes without error.
5. Check that your Networking NetView still works.
6. Start the NetView with the satellite installed and the SA OS/390 topology manager configured. At this point, the SA OS/390 topology manager should automatically contact all defined target sysplexes, retrieve their configuration information and create corresponding objects within RODM. Finally it will run the BLDVIEWS statements that you have defined for each sysplex. These will create views within RODM allowing you to see the objects created by the SA OS/390 topology manager.
7. Start an NMC server connected to the focal point system and then connect to it from an NMC client. You should see the views defined by your BLDVIEW statements. These should contain objects representing the automated resources on the target sysplexes. There should be a green heartbeat icon for each active target sysplex.
8. If you select an icon representing an automated resource and right-click, you should see SA OS/390 commands on its context menu. Select INGINFO and see that the command is issued properly.
9. Shut down the new Networking NetView, bring up the former one, and plan for production cutover.

Customizing Host System Components

1. Customize NetView DSIPARM concatenation

   Note: Most of the necessary customization of the NetView DSIPARM concatenation has already been done as a result of the concatenation of an additional SA OS/390-specific data set to the DSIPARM concatenation.

   Make the described changes to the following members in a NetView DSIPARM concatenation:

   AOFRODM

   Copy and edit the AOFRODM member, which resides in ING.SINGNPRM, as follows:

- Specify a RODM name by changing RODMNAME=NONE to RODMNAME=xxxxxxxx, where *xxxxxxxx* is your RODM name.
- Specify a user ID by changing RODMUSER=XXAOCFR to RODMUSER=xxxxxxxx, where *xxxxxxxx* is your user ID for batch updates from NetView.

2. Customize the SYS1.PARMLIB data set.

IEAAPFxx
Customization of the IEAAPFxx member should have been done during installation of the SA OS/390 automation base.

Resident Automation Procedures

SA OS/390 has a large number of NetView REXX automation procedures (also known as REXX command lists or REXX CLISTs). The customization dialog's ***Resident CLIST Definitions*** panel can be used to specify which automation procedures are to be made storage resident. The choice of which automation procedures should be made resident is important because of the performance improvements that can be gained in a trade-off against the use of storage.

The following is a list of available groups of CLISTS, with indications as to when to make them resident.

CLIST Group Name	Purpose
AOF_BASIC	System operations basic routines; should always be resident
AOF_CMD_SLAVE	System operations command slaves; should be resident if you intensively use the SA OS/390 commands (as described in the *System Automation for OS/390 Operator's Commands*).
AOF_COMMON	System operations common routines (APIs); should be resident if you use your own code which exploits these common routines documented in *System Automation for OS/390 Programmer's Reference*.
AOF_GENERIC	System operations generic routines; should always be resident because they are internally used by SA OS/390

These groups of CLISTS are included in the SA OS/390 sample policy databases. You can view these groups by creating a new policy database from one of the sample policy databases and then selecting option 41 (Resident CLISTs) from panel *SA OS/390 Entry Type Selection* in the customization dialog.

These recommended automation procedures should be considered a starting point. You should check the usage of automation procedures and add or delete them from the resident list as appropriate.

Enabling SA OS/390 Support for Extended Multiple Console Support (EMCS)

This section describes how to set up extended multiple console support and also describes EMCS's restrictions and limitations.

Note: EMCS support is mandatory for the successful operation of SA OS/390.

Setting Up EMCS

- Assign each CSSIR task a unique TSKID name within the sysplex. To do this, CNMCSSIR needs to be replaced with a unique name in DSIDMN for supported NetView below 1.4, and in DSIMSG00 and DSIMSG01 for all supported NetView.
- Update the synonym %AOFSIRTASK%, in member AOFMSGSY, to reflect the new name of the NetView CNMCSSIR task.
- Code MSGIFAC=SYSTEM on both the NetView task (in DSIDMNK) and the SSI task (in the procedure itself).
- Add the AOCGETCN command to the initial CLIST of your operator profiles.
- Switch on the SA OS/390 global variable AOF_EMCS_AUTOTASK_ASSIGNMENT, to assign an autotask to EMCS consoles.

EMCS Restrictions and Limitations

- You must code MSGIFAC=SYSTEM on both the NetView and SSI tasks.
- There must be only one NetView running SA OS/390 in each machine.
- Do not:
  - Use route codes to route messages to any NetView task console
  - Deactivate the action message retention facility (AMRF) (by coding COM=′K M,AMRF=N′ in the COMMNDxx member of SYS1.PARMLIB)
  - Change the MSCOPE setting on the xxxCSSIR task/console
  - Define the AUTO attribute for any NetView task/console under the RACF OPERPARMS
  - Define a SAF OPERPARM definition for extended MCS console authority to anything other than MASTER

Violation of these restrictions will cause unpredictable results.

Appendix D. Processor Operations Sample

This section provides a sample how to use processor operations for a NetView connection:

Host VTAM Definitions for a NetView Connection through an OSA Adapter

Channel attached Major node:

```
*=====================================================================*
*    VTAM XCA MAJNODE OVER OSA                                        *
*=====================================================================*
IPSLXCA1 VBUILD TYPE=XCA
IPSLPCA1 PORT  ADAPNO=1,                                               +
               CUADDR=110E,                                            +
               MEDIUM=RING,                                            +
               SAPADDR=04,                                             +
               TIMER=60
*** ___________________________________________________________________
***
IPSLXTG1 GROUP DIAL=YES,                                               +
               DYNPU=YES,                                              +
               DYNPUPFX=PX,                                            +
               ANSWER=ON,                                              +
               AUTOGEN=(16,L,P),                                       +
               CALL=INOUT,                                             +
               ISTATUS=ACTIVE
```

Switched Major Node for a processor operations support element (SE)

```
*=====================================================================*
*    VTAM SW MAJNODE                                                  *
*=====================================================================*
***
IPSLSWN2 VBUILD TYPE=SWNET,MAXNO=1,MAXGRP=1
*** ___________________________________________________________________
***

IPSL1T00 PU    ADDR=C1,                                                +
               IDBLK=05D,                                              +
               IDNUM=E0000,                                            +
               CPNAME=IPSL1T00,                                        +
               MAXOUT=7,                                               +
               MAXPATH=1,                                              +
               MAXDATA=265,                                            +
               PUTYPE=2,                                               +
               DISCNT=NO,                                              +
               VPACING=0,                                              +
               PACING=0,                                               +
               IRETRY=YES,                                             +
               PASSLIM=1,                                              +
               ISTATUS=ACTIVE,                                         +
               MODETAB=AMODETAB,                                       +
               DLOGMOD=D4A32782,                                       +
               USSTAB=USSCP,                                           +
               SSCPFM=USSSCS
IPSLT0   PATH  DIALNO=010400203529D9CC,GRPNM=IPSLXTG1,CALL=INOUT
*** ___________________________________________________________________
***
IPSLT000 LU    LOCADDR=0,DLOGMOD=#INTER                ProcOps localLU
```

```
*** ___________________________________________________________________
***
IPSL1T01 PU    ADDR=C1,                                                +
               IDBLK=05D,                                              +
               IDNUM=D0000,                                            +
               CPNAME=IPSL1T01,                                        +
               MAXOUT=7,                                               +
               MAXPATH=1,                                              +
               MAXDATA=265,                                            +
               PUTYPE=2,                                               +
               DISCNT=NO,                                              +
               VPACING=0,                                              +
               PACING=0,                                               +
               IRETRY=YES,                                             +
               PASSLIM=1,                                              +
               ISTATUS=ACTIVE,                                         +
               MODETAB=SNAMODET,                                       +
               DLOGMOD=D4A3278A,                                       +
               USSTAB=USSCP,                                           +
               SSCPFM=USSSCS
IPSLT1   PATH  DIALNO=010400203529D9CC,GRPNM=IPSLXTG1,CALL=INOUT
*** ___________________________________________________________________
***
IPSMT100 LU    LOCADDR=0,MODETAB=MTLU6,DLOGMOD=#INTER   Support Element
IPSMT101 LU    LOCADDR=2                                3270 SESSION
IPSMT102 LU    LOCADDR=3                                3270 SESSION
```

Appendix E. Migration Information

This appendix provides information about aspects of migration from earlier releases.

Migrating to SA OS/390 2.2

This section describes actions you must take to migrate to SA OS/390 2.2.

Migrating to SA OS/390 2.2 from SA OS/390 1.3

> **Note:**
> SA OS/390 2.2 does not accept manually edited automation control files.

The following chapter outlines the required steps to migrate to SA OS/390 2.2.

1. SA OS/390 2.2 requires that the automation control file and the associated automation manager configuration file are consistent and built from the policy database in one step.
   a. Take a copy of the current policy database.
   b. Edit the policy database with the SA OS/390 2.2 customization dialog. This will convert it to the SA OS/390 2.2 format.
   c. Migrate any user include files into the current policy database. For details, see *System Automation for OS/390 Defining Automation Policy*.
   d. The following automated functions are mandatory in order to successfully initialize SA OS/390 2.2:
      ```
      BASEOPER
      SYSOPER
      RECOPER
      MSGOPER
      MONOPER
      JESOPER
      NETOPER
      GSSOPER
      RPCOPER
      SHUTOPER
      LOGOPER
      XCFOPER
      XCFOPER2
      ```
      Some of them have already been added for SA OS/390 1.3. If you want to use GATEWAY connections to other systems you also have to have GATOPER defined. If you want to use the hardware interface functions, you must also define HWOPER01 and HWOPER02.
   e. As SA OS/390 2.2 no longer supports I/O exception monitoring, you must do the following:
      1) Exclude message table fragment AOFMSGIO from your NetView message table. This fragment contains all calls for I/O exception monitoring.
      2) Delete the following applications from your policy database:
         - IO_HEARTBEAT_TASK
         - IO_MON_EVENTPROC

- IO_MON_MSGPROC
- IO_MON_QUERYPROC

3) Delete the following automation operator definitions from your policy database:
   - INGOP1
   - INGOP2
   - INGOP3

f. If you want to use SA OS/390 for product automation (formerly called ***Features***), and you want to reuse existing automation control files, then you need to perform certain actions to migrate and convert these automation control files. Information on the required migration and conversion and how it is supported by SA OS/390 2.2 is provided in *System Automation for OS/390 Defining Automation Policy*.

   After these automation control files are converted to the SA OS/390 2.2 policy database format, you need to remove the environment setup user exits from the policy database. You achieve this by editing the policy item AUTOMATION SETUP for entry type *System*.

   Note: Groups in IMS could contain subsystems that run on different sysplexes. This is no longer possible in SA OS/390 2.2 grouping support.

g. With the introduction of the automation manager configuration file, there are four new members per sysplex and stand alone system in the automation control file build output data set. Dependent on the configuration data in the policy database, the size of the automation control file build output data set needs to be increased.

h. Build the system operations control files (automation control file and automation manager configuration file) from the customization dialog. For more information, see *System Automation for OS/390 Defining Automation Policy*.

2. Delete and re-allocate the automation status file (ASF). Be aware that you will loose:
   - service period overrides
   - current settings of events
   - your user data that you may have stored in the ASF via the ASFUSER command
3. If you have installed the special program enhancement (SPE) ***SA OS/390 Sysplex Automation*** (APAR OW39485/PTF UW99278), you need to migrate your policy for automating message IXC102A as delivered with this SPE for SA OS/390 1.3 to the new SA OS/390 format. Refer to the documentation in *System Automation for OS/390 Customizing and Programming* and adapt your automation policy as described.

Migrating to SA OS/390 2.2 from SA OS/390 2.1

The following steps are required to migrate to SA OS/390 2.2.

ACF Migration

SA OS/390 2.2 requires that the automation control file and the associated automation manager configuration file are consistent and built from the policy database in one step.

1. Take a copy of the current policy database.

2. Edit the policy database with the customization dialog. This will convert it to the SA OS/390 2.2 format.
3. Migrate any user include files into the current policy database. For details, see *System Automation for OS/390 Defining Automation Policy*.
4. Change MAT AOFMSG01 to INGMSG01 in the AUTOMATION SETUP of system policy object.
5. Build the system operations control files (automation control file and automation manager configuration file) from the customization dialog. For more information, see *System Automation for OS/390 Defining Automation Policy*.

NMC Migration

Delete the following files on the NMC Topology Server:

```
files             directory
----------        ---------------------
ING_NMCC_*.DDF    \usr\local\Tivoli\bin\w32-ix86\tds\server\config\ddf\
ISQ_NMCC_*.DDF    \usr\local\Tivoli\bin\w32-ix86\tds\server\config\ddf\
ING_NMCC_*.HTML   \usr\local\Tivoli\bin\w32-ix86\tds\server\db\current\help\
ISQ_NMCC_*.HTML   \usr\local\Tivoli\bin\w32-ix86\tds\server\db\current\help\
```

The following files used in SA OS/390 2.1 are replaced by files shipped with SA OS/390 2.2:

```
INGNMCJDial.jar
ING_NMCS_CMD.RSP
ISQ_NMCS_CMD.RSP
```

The files have identical names in both releases.

To apply the new response files for SA OS/390 2.2, use the same method you use for installing fixes. For a detailed description and latest information about how to apply changes, refer to the SA OS/390 Web site at:
`http://www.ibm.com/servers/eserver/zseries/software/sa/adds/hint03.html`

Coexistence with SA OS/390 1.3 and SA OS/390 2.1

It is not expected that a customer will cut over all their systems at the same time from SA OS/390 1.3 to SA OS/390 2.2, or from SA OS/390 2.1 to SA OS/390 2.2. This means, that you are running two different releases at the same time.

The following outlines the support that SA OS/390 provides for such an environment:

- You do **not** need to maintain different versions of the policy database. Instead, the customization dialog of SA OS/390 2.2 can be used to maintain the policy database. The ACF built by the customization dialog can be used by a down-level SA OS/390 release but only when no new SA OS/390 2.2 function has been selected. Refer to Chapter 2, “What is New in SA OS/390 2.2”, on page 9 for more information on new SA OS/390 2.2 functions.
- SA OS/390 2.2 systems can co-exist with SA OS/390 1.3 and SA OS/390 2.1 systems within the same sysplex. Figure 34 on page 186 illustrates this: It shows a sysplex with six automated systems and a separate automation manager.

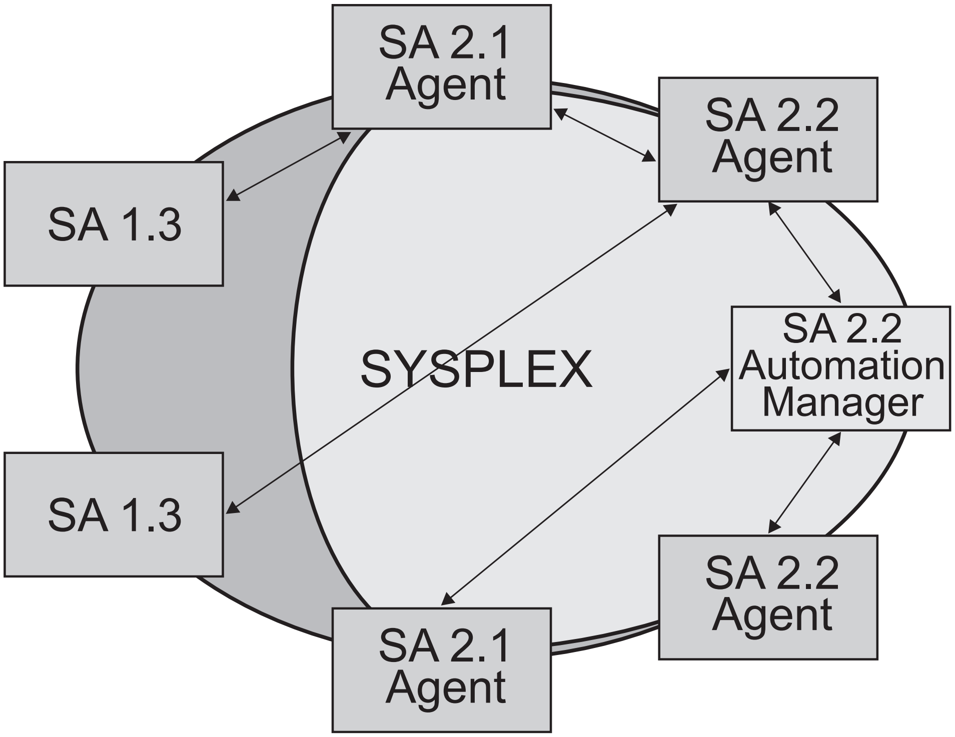


Figure 34. Coexistence of SA OS/390 1.3, SA OS/390 2.1, and SA OS/390 2.2

The SA OS/390 1.3 systems and the SA OS/390 automation agents register at initialization time, so that each system has knowledge about the resources that run on each of the other systems. This allows the operator on a SA OS/390 1.3 system to control resources (subsystems) that run on a SA OS/390 2.2 system. However, the SA OS/390 1.3 system will not communicate with the SA OS/390 2.2 automation manager. If you are running SA OS/390 2.1 and SA OS/390 2.2, the SA OS/390 2.2 automation manager must be started. The SA OS/390 2.2 automation manager communicates with the automation agents of SA OS/390 2.1 and SA OS/390 2.2.

There are some other restrictions concerning SA OS/390 commands in such an environment. These are described in *System Automation for OS/390 User's Guide*.

NMC exploitation is only supported between SA OS/390 2.1 and SA OS/390 2.2.

SA OS/390 2.1 and SA OS/390 2.2 do not communicate with SA OS/390 1.2.

Coexistence of Service Periods, Events and Triggers with SA OS/390 1.3

Service periods linked to triggers (from SA OS/390 1.3) will not be deleted but will be ignored by SA OS/390 2.2. With the first opening of a policy database from SA OS/390 1.3, SA OS/390 2.2 will automatically link the service periods directly to the applications or application groups to which the trigger has been linked in the SA OS/390 1.3 policy database. For more information refer to *System Automation for OS/390 Defining Automation Policy*.

For reasons of downward compatibility with SA OS/390 1.3 the build functions will build ACF fragments for service periods, events and triggers. These will be ignored by SA OS/390 2.2, as information for these entry types are taken from the AMC file.

For SA OS/390 1.3, if a trigger has a service period which is not referenced in at least one startup and one shutdown condition, the *BUILD* command of the customization dialog generates an ACF entry with one additional startup and/or one additional shutdown condition with the value `SERVICE`.

Note:
Due to the different presentation and the different behaviour of the new service period overrides, the overrides from SA OS/390 1.3 cannot be migrated to SA OS/390 2.2.

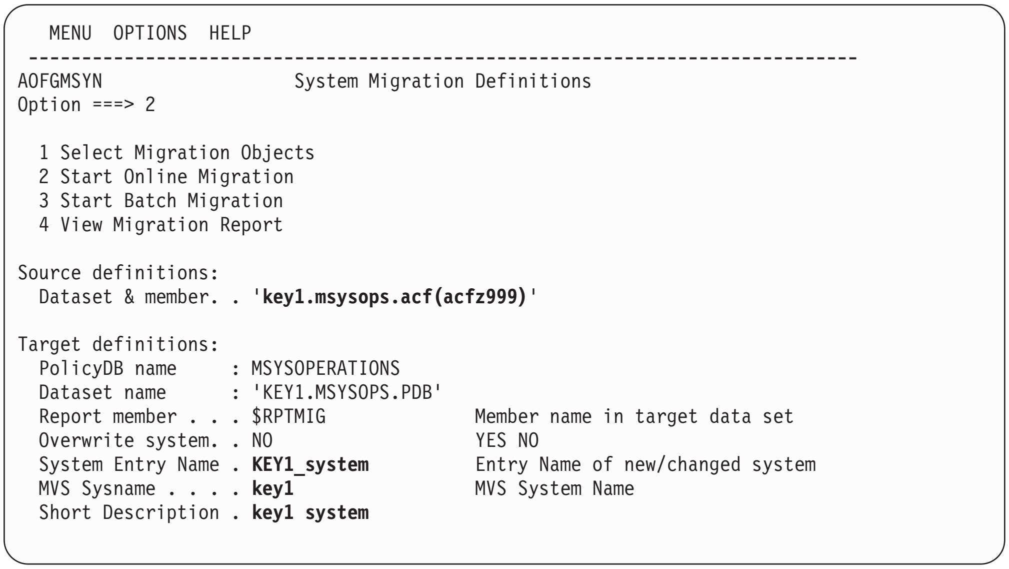
Migrating to SA OS/390 2.2 from msys for Operations

If you are using msys for Operations and you are going to migrate to SA OS/390 2.2, you may migrate system by system. The msys for Operations functions will cooperate with SA OS/390 2.2, if the same XCF group ID is used. You can achieve this by setting GRPID in the INGXINIT member (for the automation agents) and in HSAPRM*xx* (for the automation manager) to `A0` (’A’, zero) during the migration phase.

To migrate from msys for Operations to System Automation for OS/390, you must perform the following steps:

1. Log on to msys for Operations and use the command INGCUST to migrate the entries of member AOFCUST to ACF fragments. Refer to “INGCUST” on page 188 for details.
2. Use the SA OS/390 customization dialog to create a new policy database by selecting a sample database as a model (sample DEFAULTS is recommended). For detailed information, refer to the chapter that describes how to use the customization dialog in *System Automation for OS/390 Defining Automation Policy*.
3. Use the MIGRATE function of the SA OS/390 customization dialog to add the automation definitions to the policy database. For details, refer to ″Migration Functions″ in *System Automation for OS/390 Defining Automation Policy*.

   On the System Migration Definitions panel (AOFGMSYN), enter the *pds_library_name* you specified in INGCUST and member ACFZ999. MVS Sysname must be the name of a system in the sysplex. See Figure 35 on page 188 for an example.



```
   MENU  OPTIONS  HELP
 ------------------------------------------------------------------------------
AOFGMSYN                 System Migration Definitions
Option ===> 2

  1 Select Migration Objects
  2 Start Online Migration
  3 Start Batch Migration
  4 View Migration Report

Source definitions:
  Dataset & member. . 'key1.msysops.acf(acfz999)'

Target definitions:
  PolicyDB name     : MSYSOPERATIONS
  Dataset name      : 'KEY1.MSYSOPS.PDB'
  Report member . . . $RPTMIG              Member name in target data set
  Overwrite system. . NO                   YES NO
  System Entry Name . KEY1_system          Entry Name of new/changed system
  MVS Sysname . . . . key1                 MVS System Name
  Short Description . key1 system
```

Figure 35. System Migration Definitions Panel

Start processing the migration.

4. After having migrated one system with the migration function, you must now create the other systems of the sysplex (use entry type SYS of the customization dialog), and copy all entries from the migrated system to the new defined systems by using policy item COPY.
5. Make sure that all systems are linked to the sysplex group.
6. Customize your SA OS/390 according to your requirements.

INGCUST

INGCUST is a service function that you can use to migrate from msys for Operations to the licensed version of System Automation for OS/390. It converts the sections of AOFCUST to ACF fragments that must be stored in a PDS library.

Specify INGCUST with the partitioned data set (PDS) name, where the ACF fragments will be stored.

```
►►──INGCUST──pds_library_name──────────────────────────────────────────────►◄
```

pds_library_name must be allocated with a fixed-block format (RECFM=FB), and a logical record length of 80 (LRECL=80). It also requires 3 directory blocks and 2 tracks.

The ACF fragments will be built and stored as PDS members within *pds_library_name*.

Note: The NetView operator must have write-access to this partitioned data set.

Appendix F. Syntax for HSAPRM00

Notes:

1. A sample member called HSAPRM00 is provided in the SINGFSAMP sample library.
2. Records starting with a ′*′ in column 1 are treated as comments. Each parameter must be specified on a single line. Trailing comments are not supported.

```
BUILDTIMEOUT={ss | 180}
CFGDSN=<configuration file data set name>
COMM={XCF | MQ}
DELAY={ss | 0}
DIAGINFO=dsname
GRPID={xx | '  '}
LEOPT={<any>}
MQM=ssid
NUMQTHDS={n | 3}
OVRDELETEDELAY={dd | 0}
PREF=number
PROMPT={YES | NO}
START={COLD | HOT | WARM}
STOPDELAY={ss | 30}
TAKEOVERTIMEOUT=nn
```

BUILDTIMEOUT

may be used to specify a time limit for the completion of the data structure build process as used during COLD or WARM start of the primary automation manager. A value from 0-999 seconds may be specified, and a value of 180 (3 minutes) will be assumed if omitted. A specification of 0 suppresses timing of the data structure build process.

CFGDSN

The CFGDSN value will be used only on a COLD start, and may be overridden by an initialization prompt response. On other start types, the default CFGDSN will be that in use when automation was last active. No default CFGDSN value is provided, and will result in a prompt for the data set to be used.

COMM

This parameter specifies how communication between the automation manager and the automation agents is realized. The possible values are:

XCF specifies that the automation manager will use XCF for communication with the automation agents. In this case, the takeover file provides the persistent storage medium for holding the current resource states and settings across automation manager sessions.

Using XCF for communication has the following risks:

- All workitems travelling to, queued in, or processed by the automation manager are lost when the automation manager terminates abnormally.
- Orders for the automation agents can be broken because some orders could already have been sent at the time when the automation manager terminated abnormally.
- A warmstart is required when an irrecoverable I/O error occurs while reading from or writing to the takeover file.

MQ This is the recommended option. It specifies that the automation manager will use MQSeries for communication with the automation agents, and also for holding the status information. With this option, the information in the header of the takeover file determines whether MQSeries or the takeover file is used for a HOT start or takeover.

The COMM parameter and the MQ parameter are mutually dependent. When you specify `COMM=XCF`, the MQ parameter must be left blank. With `COMM=MQ` you must specify an MQSeries subsystem for the MQ parameter.

DELAY
is the number of seconds to be used as a default delay prior to determining the operational mode at when the automation manager instance is started. This value may by overridden on an individual instance basis by the start command parameter. A delay value from 0-999 seconds may be specified, and value of 0 (no delay) will be assumed if omitted. This parameter will be ignored when the automation manager instance is started by Automatic Restart Manager or with the specification of TYPE=HOT.

DIAGINFO
specifies that the automation manager starts workitem recording from the beginning. `dsname` is the name of the data set that will hold the workitems. The data set must be a sequential file. It must exist and must be catalogued.

Note: The data set name is accepted without checking if the data set exists or if it is accessed by another user.

GRPID
is the same as that currently generated by the SA OS/390 1.3 customization dialog, and will be prefixed with the string INGXSG to create the XCF group name as used by the communication manager function. Note that this value cannot be overridden, and that a null value will be used if not specified.

LEOPT
may be used to pass run-time options to the LE environment.

1. Options forced by the Automation Manager

   The following LE runtime options are set by the Automation Manager during initialization:

   ALL31(ON)
   POSIX(ON)

   Note: These options must not be overridden by installation default settings (CEEDOPT) with the NONOVR attribute.

2. Default options set by the Automation Manager during initialization

   The following LE runtime options are set by the Automation Manager during initialization:

   ```
   ANYHEAP(2m,256K,ANYWHERE,FREE)
   DEPTHCONDLMT(4)
   ERRCOUNT(0)
   HEAP(6M,1M,ANYWHERE)
   STACK(64K,64K,ANYWHERE,KEEP)
   STORAGE(NONE,NONE,NONE,128K)
   ```

Note: These options may be overidden by the customer.

3. Recommended LE Options

   The following LE options are recommended for the System Automation Manager:

   ```
      NONIPTSTACK(4K,4K,ANYWHERE,KEEP)
   or THREADSTACK(ON,4K,4K,ANYWHERE,KEEP,512K,128K)
             Note:  NONIPTSTACK was replaced by THREADSTACK in OS/390 LE 2.10
      PROFILE(OFF,'')
      RTLS(OFF)
      STORAGE(NONE,NONE,NONE,128K)
      THREADHEAP(4K,4K,ANYWHERE,KEEP)
      TRACE(OFF,4K,DUMP,LE=0)
      VCTRSAVE(OFF)
      XPLINK(OFF)
   ```

   The following LE options should be tuned using the LE storage reporting facility RPTSTG(ON):

   ```
   ANYHEAP(2M,256K,ANYWHERE,FREE)
   HEAP(6M,1M,ANYWHERE,KEEP)
   HEAPPOOLS(ON,40,2,64,2,104,2,312,2,624,1,2024,1)
   STACK(64K,64K,ANYWHERE,KEEP)
   ```

   The following option is used to direct output created as a result of specifying RPTOPTS(ON) or RPTSTG(ON). It is also used to direct diagnostic messages written to CEEMSG and CEEMOUT by the Automation Manager.

   ```
   MSGFILE(SYSOUT,FBA,121,0,NOENQ)
   ```

   The following options are recommended for OS/390 LE Version 2 Release 10:

   ```
   HEAP(6M,1M,ANYWHERE,KEEP,32K,16K)
   THREADSTACK(ON,4K,4K,ANYWHERE,KEEP,512K,128K)
   ```

   The storage options for the below the line heap need to be tuned.

   The following options can be used to gather diagnostic and storage-usage information:

   ```
    RPTSTG(ON)
    RPTOPTS(ON)
   ```

Notes:

1. If an LEOPT= keyword is present in HSAPRM00, it replaces any LEOPT that may have been specified as an input parameter through JCL.
2. When specifying options in HSAPRMxx you may have LEOPT statements on a number of different lines, but the total length of all of the options cannot exceed 4096 characters.

Sample LEOPTS statements are supplied in sample member HSAPRM00.

MQM This value specifies the subsystem ID (SSID) of the current MQSeries manager.

The MQ parameter and the COMM parameter are mutually dependent. When you specify `COMM=XCF`, the MQ parameter must be left blank. With `COMM=MQ` you must specify an MQSeries subsystem for the MQ parameter.

NUMQTHDS

The NUMQTHDS parameter controls the number of query threads. This value limits the amount of parallel query activity that can be performed. If not specified, a default value of 3 will be used. A maximum of 15 query threads may be specified.

OVRDELETEDELAY
is the number of days that a schedule override should be retained before being automatically deleted. A value of 0 days indicates that schedule overrides are not to be automatically deleted and is the default if no value is specified. A maximum of 366 days may be specified.

PREF specifies the preference given to the instance of the automation manager when determining which of the SAMs should become primary automation manager. The number can range from 0 to 15, where 0 is the highest preference. The SAM will only participate in the escalation process when there is no other SAM active with higher preference. The default is 0.

PROMPT
Specifying YES lets you overwrite the *CFGDSN* parameter (the name of the automation manager configuration file). Message HSAM1302A will come up and wait for a response. You may now specify the keyword/value pair

```
CFGDSN=<fully.qualified.data.set.name>
```

or you may use a null or 'U' response to indicate no override values are to be applied.

START
defines the start mode of the automation manager. During initialization, the automation manager retrieves input from:

1 parameter CFGDSN

2 schedule overrides

3 persistent data store (votes, triggers, resource states)

The following table shows where the automation manager retrieves initialization data for the possible values for parameter START.

	COLD	**WARM**	**HOT**
1	name of automation manager configuration file is taken from PARMLIB, the START command or via PROMPT=YES option	last used value taken	last used value taken
2	deleted	taken from last run	taken from last run
3	deleted	deleted	taken from last run

Recommendation: Use COLD for the very first time, or when the schedule override file should be cleared. Use WARM if the automation policy has changed, that is, the automation manager configuration file has been rebuilt. Use HOT in any other case.

The start mode does not affect the secondary automation managers.

STOPDELAY
is the number of seconds to be used when an MVS F <jobname>,STOP,DEFER command is entered for the primary automation

manager. This delay will be invoked only if one or more secondary automation managers are active and ready when the command is received.

TAKEOVERTIMEOUT

nn may range from 60 to 600 seconds.

If the (secondary) automation manager performs a takeover, or an automation manager is started HOT, then it examines whether MQSeries is ready. If this is not the case, the automation manager enters a retry-loop. The TAKEOVERTIMEOUT parameter determines how many seconds the automation manager should wait (retry) until it switches from mode=HOT to mode=WARM.

This parameter does not apply when the COMM parameter is set to XCF.

Syntax for HSAPRM00

Appendix G. Example Automation Network Configuration

Figure 36 shows an example network of four systems, all running NetView. Two systems are in Chicago, and two are in Atlanta.

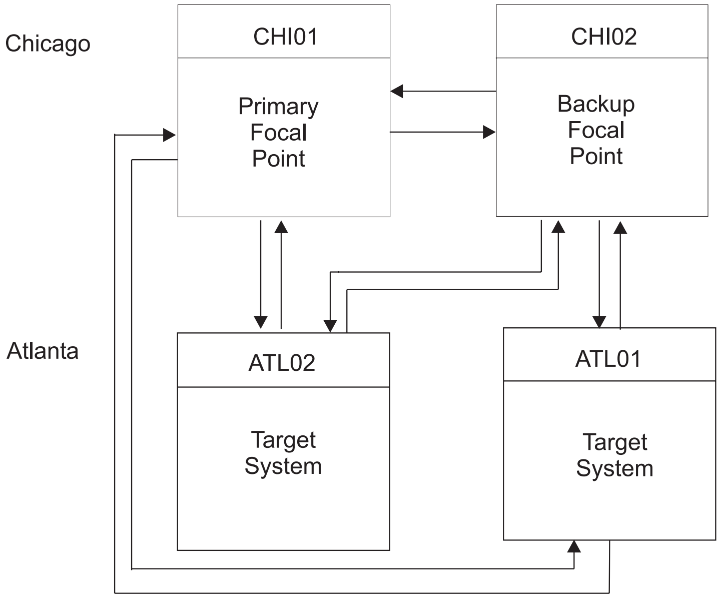


Figure 36. Example Automation Network

Following are desired configuration features of the network shown in Figure 36:

- System CHI01 is the focal point system and system CHI02 is the backup focal point system.
- The gateway session between systems CHI01 and ATL01 is used as a path from system CHI01 to systems ATL02, ATL03, and ATL04.
- The gateway session between systems CHI02 and ATL01 is used as a path from system CHI02 to systems ATL02, ATL03, and ATL04.
- If the gateway session between systems CHI01 and ATL01 is inactive, the gateway session between ATL01 and CHI02 can be used as an alternate path to CHI01.
- The SDF in focal point system CHI01 can be tailored to indicate the status of all other systems.
- All systems use the SA OS/390 password protection feature to generate and maintain gateway autotask passwords.
- Systems CHI01 and CHI02 share the same RACF data set.
- Target systems ATL02 and ATL04 include adjacent NetView definitions of system ATL03 so that commands and responses can be routed between them through system ATL03.

To configure focal point services for the example network, the following definitions are required:

- Automation policy settings (customization dialog values)
- NetView DSIPARM DSIOPF entries
- NetView DSIPARM DSIDMN entries

- RACF definitions

Customization activities for each system, including entering values in the customization dialog, building an automation control file, and modifying data sets, must be performed on that system. Details on customization activities required for each system follow.

NetView DSIPARM DSIOPF Entries for Example Automation Network

For each system in an automation network, you specify the names of all operator IDs for both inbound and outbound gateway autotasks in NetView DSIPARM data set member DSIOPF. Following are the DSIOPF entries required to configure the example network shown in Figure 36 on page 195.

We recommend to use AOFOPFGW which is included in AOFOPF to do this customization.

System CHI01:

```
GATCHI01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATCHI02 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
```

System CHI02:

```
GATCHI01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATCHI02 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
```

System ATL01:

```
GATCHI01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL02 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL03 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL04 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATCHI02 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
```

System ATL02:

```
GATATL01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL02 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL03 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
```

System ATL03:

```
GATATL01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL02 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
```

```
GATATL03 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL04 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
```

System ATL04:

```
GATATL01 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL03 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
GATATL04 OPERATOR PASSWORD=RACFNNT
         PROFILEN AOFPRFAO
```

NetView DSIPARM DSIDMN Entries for Example Automation Network

For each system in an automation network, you add or modify CDMNSESS, and RRD statements in NetView DSIPARM data set member DSIDMN to reflect the automation network configuration. Following are the entries in the NetView DSIPARM data set member DSIDMN entries required to configure the example network shown in Figure 36 on page 195.

System CHI01:

```
CDMNSESS  04
ATL01 RRD
CHI02 RRD
```

System CHI02:

```
CDMNSESS  04
CHI01 RRD
ATL01 RRD
```

System ATL01:

```
CDMNSESS  10
CHI01 RRD
ATL02 RRD
ATL03 RRD
ATL04 RRD
CHI02 RRD
```

System ATL02:

```
CDMNSESS  04
ATL01 RRD
ATL03 RRD
```

System ATL03:

```
CDMNSESS  06
ATL01 RRD
ATL02 RRD
ATL04 RRD
```

System ATL04:

```
CDMNSESS  04
ATL01 RRD
ATL03 RRD
```

Password Definitions for Example Automation Network

Following are the password definitions in the SA OS/390 password and RACF data sets required to configure the example network shown in Figure 36 on page 195.

Note: This example assumes the SA OS/390 password protection feature has been defined on all systems, as described in *System Automation for OS/390 Planning and Installation*.

The SA OS/390 password data set for system CHI01 contains passwords for gateway operator IDs named GATCHI01 on systems CHI02 and ATL01.

The SA OS/390 password data set for system CHI02 contains passwords for gateway operator IDs named GATCHI02 on systems CHI01 and ATL01.

The SA OS/390 password data set for system ATL01 contains a single password entry for logging gateway operator ID GATATL01 on to systems CHI01 and CHI02. There is only one entry for both systems because CHI01 and CHI02 share the same SAF data set. The password data set also contains entries for logging a gateway operator ID named GATATL01 on to systems ATL02, ATL03, and ATL04.

The SA OS/390 password data set for system ATL02 must contain passwords for logging an operator ID named GATATL02 onto systems ATL01 and ATL03.

The SA OS/390 password data set for system ATL03 must contain passwords for logging an operator ID named GATATL03 onto systems ATL01, ATL02, and ATL04.

The SA OS/390 password data set for system ATL04 must contain passwords for logging an operator ID named GATATL04 onto systems ATL01 and ATL03.

On each system, the INGPW command must be issued to initialize the VSAM SA OS/390 password data set with the passwords for gateway operator IDs. The password issued on the INGPW command is the initial password value identified for the operator ID by the your security administrator. The INGPW commands issued for each system are as follows (in these commands the `password` field contains the actual password for the outbound gateway operator to log on to the specified system):

System CHI01:

```
INGPW GATCHI01 CHI02,INIT=password
INGPW GATCHI01 ATL01,INIT=password
```

System CHI02:

```
INGPW GATCHI02 CHI01,INIT=password
INGPW GATCHI02 ATL01,INIT=password
```

System ATL01:

```
INGPW GATATL01 CHI01,INIT=password
INGPW GATATL01 ATL02,INIT=password
INGPW GATATL01 ATL03,INIT=password
INGPW GATATL01 ATL04,INIT=password
```

System ATL02:

```
INGPW GATATL02 ATL01,INIT=password
```

System ATL03:

```
INGPW GATATL03 ATL01,INIT=password
```

System ATL04:

```
INGPW GATATL04 ATL01,INIT=password
```

Glossary

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing* New York: McGraw-Hill, 1994.
- The *American National Standard Dictionary for Information Systems* , ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.
Deprecated term for. This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.
See. This refers the reader to multiple-word terms in which this term appears.
See also. This refers the reader to terms that have a related, but not synonymous, meaning.
Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.
Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

A

ACF. Automation control file.

ACF/NCP. Advanced Communications Function for the Network Control Program. See *Advanced Communications Function* and *Network Control Program*.

ACF/VTAM*. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*. See *Advanced Communications Function* and *Virtual Telecommunications Access Method*.

ACO. Automated console operations.

active monitoring. In SA OS/390, 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 OS/390, the system defined as the communication path between two SA OS/390 systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA OS/390 systems. For example, the adjacent NetView is used when sending responses from a focal point system to a remote system.

Advanced Communications Function (ACF). A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

advanced program-to-program communication (APPC). A set of inter-program communication services that support cooperative transaction processing in a Systems Network Architecture (SNA) network. APPC is the implementation, on a given system, of SNA's logical unit type 6.2.

alert. (1) In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition. (2) In the NetView program, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

alert condition. A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.

alert focal point system. See entry for NPDA focal point system under *focal point system*

alert threshold. An application service value that determines the level at which SA OS/390 changes the associated icon on the NMC workstation to the alert color. SA OS/390 may also issue an alert. See *warning threshold*

AMC. (1) Automation Manager Configuration (2) The Auto Msg Classes entry type

APF. Authorized program facility.

APPC. Advanced program-to-program communications.

application. An OS/390 subsystem or job monitored by SA OS/390.

Application entry. A construct, created with the customization dialog, used to represent and contain policy for an application.

application group. A named set of applications. An application group is used to represent and contain common policy for the contained applications. Applications in an application group can be automated as an entity. Application groups can behave differently according to their defined nature (BASIC, MOVE, SERVER). An application group can also be used for monitoring purposes.

ApplicationGroup entry type. A construct, created with the customization dialog, used to represent and contain policy for an application group.

application program. (1) A program written for or by a user that applies to the user's work, such as a program that does inventory or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

ARM. Automatic restart management.

ASCB. Address space control block.

ASCB status. An application status derived by SA OS/390 running a routine (the ASCB checker) which searches the OS/390 address space control blocks (ASCBs) for address spaces with a particular job name. The job name used by the ASCB checker is the job name defined in the customization dialog for the application.

ASCII (American National Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8-bits including parity check), for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

ASF. automation status file.

assist mode facility. An SA OS/390 facility that uses SDF and enables interaction with automation before SA OS/390 takes an automation action. SDF prompts the operator with a suggested action, then provides options for using that action, modifying and using the action, or canceling the action. Also called assist mode, it is enabled using the customization dialog, or dynamically.

authorized program facility (APF). A facility that permits identification of programs that are authorized to use restricted functions.

automated function. SA OS/390 automated functions are NetView automation operators, that are assigned to perform specific automation functions. However, SA OS/390 defines its own synonyms, or *automated function names*, for the NetView automation operators, and these function names are referred to in the sample policy databases provided by SA OS/390. For example, the automation operator AUTBASE corresponds to the SA OS/390 automated function BASEOPER.

automated console operations (ACO). The concept (versus a product) of using computers to perform a large subset of tasks ordinarily performed by operators, or assisting operators in performing these tasks.

automatic restart management. An OS/390 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 OS/390.

automatic restart management element name. In MVS 5.2 or later, OS/390 automatic restart management requires the specification of a unique sixteen character name for each address space which registers with it. All automatic restart management policy is defined in terms of the element name, including SA OS/390's interface with it.

automation. The automatic initiation of actions in response to detected conditions or events. SA OS/390 provides automation for OS/390 applications, OS/390 components, and remote systems that run OS/390. SA OS/390 also provides tools that can be used to develop additional automation.

automation agent. In SA OS/390, 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 control file (ACF). In SA OS/390, a file that contains system-level automation policy information. There is one master automation control file for each NetView system on which SA OS/390 is installed. The SA OS/390 customization dialog must be used to build the automation control files. They must not be edited manually.

automation flags. In SA OS/390, 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 OS/390 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 dialog.

automation manager. In SA OS/390, the automation function is split up between the automation manager and teh 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 stand-alone system.

Automation NetView. In SA OS/390 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 OS/390 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, OS/390 subsystems, OS/390 data sets, and OS/390 components.

automation policy settings. The automation policy information contained in the automation control file. This information is entered using the customization dialog. You can display or modify these settings using the customization dialog.

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 the NetView program.

automation status file. In SA OS/390, a file containing status information for each automated subsystem, component or data set. This information is used by SA OS/390 automation when taking action or when determining what action to take.

automation table. Also called NetView message automation table. See *NetView message automation table*

autotask. See *automation operator*.

available. In VTAM programs, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

B

basic mode. A central processor mode that does not use logical partitioning. Contrast with *logically partitioned (LPAR) mode*.

BCP Internal Interface. Processor function of CMOS-390, zSeries processor families. It allows the communication between basic control programs such as z/OS and the processor support element in order to exchange information or to perform processor control functions. Programs using this function can perform hardware operations such as ACTIVATE or SYSTEM RESET.

beaconing. The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

C

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

central processor complex (CPC). A physical collection of hardware that consists of central storage, one or more central processors, timers, and channels.

central site. In a distributed data processing network, the central site is usually defined as the focal point for alerts, application design, and remote system management tasks such as problem management.

CFR/CFS and ISC/ISR. I/O operations can display and return data about integrated system channels (ISC) connected to a coupling facility and coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

channel. A path along which signals can be sent; for example, data channel, output channel. See also *link*.

channel path identifier. A system-unique value assigned to each channel path

CHPID. In SA OS/390, channel path ID; the address of a channel.

CHPID port. A label that describes the system name, logical partitions, and channel paths.

channel-attached. (1) Attached directly by I/O channels to a host processor (for example, a channel-attached device). (2) Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

CI. Console integration.

CICS/VS. Customer Information Control System for Virtual Storage.

CLIST. Command list.

clone. A set of definitions for application instances which are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

clone ID. A generic means of handling system-specific values such as the MVS SYSCLONE or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

CNC. A channel path that transfers data between a host system image and an ESCON control unit. It can be point-to-point or switchable.

command. A request for the performance of an operation or the execution of a particular program.

command facility. The component of the NetView program that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

command list (CLIST). (1) A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled. (2) In SA OS/390, REXX command lists that can be used for automation procedures.

command procedure. In the NetView program, 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 OS/390 subset of the CPC operations management commands.

common routine. One of several SA OS/390 programs that perform frequently used automation functions. Common routines can be used to create new automation procedures.

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*.

communications path. A set of communications facilities that SA OS/390 uses to connect a focal point system to a target system. There are two types: a NetView connection and a workstation.

communications task. An SA OS/390 task responsible for all communications with a number of workstations. Communications tasks receive commands from target control tasks and send inbound messages to the message monitoring task. There can be many communications tasks. Communications tasks are defined using the configuration dialogs.

configuration dialogs. A user interface for entering the information that defines the SA OS/390 configuration. In SA OS/390 the configuration dialogs are an ISPF application.

connectivity view. In SA OS/390, 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 OS/390, 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 which if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.

consoles. Workstations and 3270-type devices that manage your enterprise.

Control units. Hardware units that control I/O operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports.

controller. A unit that controls I/O operations for one or more devices.

couple data set. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the OS/390 systems in a sysplex. See also *sysplex couple data set* and *XCF couple data set*

coupling facility. The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

CP. Central processor.

CPC. Central processor complex.

CPC operations management commands. A set of commands and responses for controlling the operation of System/390 CPCs.

CPC subset. All or part of a CPC. It contains the minimum *resource* to support a single control program.

CPCB. Command processor control block; an I/O operations internal control block that contains information about the command being processed.

CPU. Central processing unit. Deprecated term for *processor*.

cross-system coupling facility (XCF). XCF is a component of OS/390 that provides functions to support cooperation between authorized programs running within a sysplex.

CTC. The channel-to-channel (CTC) channel can communicate with a CTC on another host for intersystem communication.

Customer Information Control System (CICS). A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

customization dialog. The customization dialog is an ISPF application. They are used to customize the enterprise policy, like for example the automated enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise.

CVC. A channel operating in converted (CVC) mode transfers data in blocks and a CBY channel path transfers data in bytes. Converted CVC or CBY channel paths can communicate with a parallel control unit. This resembles a point-to-point parallel path and dedicated connection, regardless whether it passes through a switch.

D

DASD. Direct access storage device.

data services task (DST). The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set members. Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

DBCS. Double-byte character set.

DCCF. Disabled console communication facility.

DCF. Document composition facility.

Devices. You can see information about all devices (such as printers, tape or disk drives, displays, or communications controllers) attached to a particular switch, and control paths and jobs to devices.

direct access storage device (DASD). A device in which the access time is effectively independent of the location of the data; for example, a disk.

disabled console communication facility (DCCF). An OS/390 component that provides limited-function console communication during system recovery situations.

disk operating system (DOS). (1) An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data. (2) Software for a personal computer that controls the processing of programs. For the IBM Personal Computer, the full name is Personal Computer Disk Operating System (PCDOS).

distribution manager. The component of the NetView program that enables the host system to use, send, and delete files and programs in a network of computers.

domain. (1) An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and associated resources that the SSCP can control by means of activation requests and deactivation requests.

double-byte character set (DBCS). A character set, such as Kanji, in which each character is represented by a 2-byte code.

DSIPARM. This file is a collection of members of NetView's DSIPARM data set. It determines how automation is applied to resources.

DST. Data Services Task.

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

ECB. Event control block. A control block used to represent the status of an event.

EMCS. Extended multiple console support.

enterprise. An organization, such as a business or a school, that uses data processing.

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 OS/390 should stop trying to restart or recover an application, subsystem or component, or off-load a data set.

ESA. Enterprise Systems Architecture.

event. (1) In the NetView program, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation. (3) Events are part of a trigger condition, in a way that if all events of a trigger condition have occurred, a *STARTUP* or *SHUTDOWN* of an application is performed.

exception condition. An occurrence on a system that is a deviation from normal operation. SA OS/390 monitoring highlights exception conditions and allows an SA OS/390 enterprise to be managed by exception.

extended recovery facility (XRF). A facility that minimizes the effect of failures in OS/390, 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*

file manager commands. A set of SA OS/390 commands that read data from or write data to the automation control file. These commands are useful in the development of automation that uses SA OS/390 facilities.

focal point. In the NetView program, 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. (1) A system that can administer, manage, or control one or more target systems. There are a number of different focal point system associated with IBM automation products. (2) **SA OS/390 automation focal point system.** The SA OS/390 automation focal point system is an SA OS/390 NetView system that collects status information from other SA OS/390 NetViews within your enterprise. It is supported by SA OS/390, which uses NetView NNT/OST sessions to forward information to it. (3) **SA OS/390 configuration focal point system.** The SA OS/390 configuration focal point is the system on which SA OS/390 is installed. The SA OS/390 policy databases exist on this system and are built into automation control file fragments, and processor control file on this system (if this is being done using SA OS/390). (4) **SA OS/390 workstation focal-system.** The SA OS/390 workstation focal point must be the same as the NMC focal point system. SA OS/390 code is installed on both the workstation and the NMC focal point system. The SA OS/390 workstation focal point system collects status information about systems and applications within your enterprise. It is supported by SA OS/390. (5) **NMC focal point system**. The NMC focal point system is a NetView system with an attached workstation server and LAN which gathers information about the state of the network. This focal point system uses RODM to store the data it collects in the data model. The information stored in RODM can be accessed from any LAN-connected workstation with

NetView Management Console installed. (6) **NPDA focal point system.** This is a NetView system which collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA OS/390 installed the NPDA focal point system must be the same as your NMC focal point system. The NPDA focal point system is also known as the *alert focal point system*. (7) **Status focal point system.** In NetView, the system to which STATMON, VTAM and NLDM send status information on network resources. If you have a NMC focal point, it must be on the same system as the Status focal point. (8) **SA OS/390 focal point system.** This is a NetView system that has SA OS/390 host code installed, and network connections to a number of SA OS/390 workstations. The SA OS/390 focal point system receives messages from the systems and operator consoles of the machines 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. (9) **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 the NetView program, 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 OS/390 outbound gateway operator logs onto the other NetView session without human operator intervention. Each end of a gateway session has both an inbound and outbound gateway operator.

generic alert. Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as NetView.

generic routines. In SA OS/390, a set of self-contained automation routines that can be called from the NetView message automation table, or from user-written automation procedures.

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 be of type STANDARD or SYSPLEX.

H

Hardware Management Console (HMC). A console used to monitor and control hardware such as CMOS-390 or zSeries processors.

Hardware Management Console Application (HWMCA). A direct-manipulation object-oriented graphical user interface that provides single point of control and single system image for hardware elements. HWMCA provides customer grouping support, aggregated and real-time system status using colors, consolidated hardware messages support, consolidated operating system messages support, consolidated service support, and hardware commands targeted at a single system, multiple systems, or a customer group of systems.

heartbeat. In SA OS/390, a function which monitors the validity of the status forwarding path between remote systems and the NMC focal point system, and monitors the availability of remote OS/390 systems, to ensure that status information displayed on the SA OS/390 workstation is current.

help panel. An online panel that tells you how to use a command or another aspect of a product.

hierarchy. In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

high-level language (HLL). A programming language that does not reflect the structure of any particular computer or operating system. For the NetView program, the high-level languages are PL/I and C.

HLL. High-level language.

host system. In a coupled system or distributed system environment, the system on which the facilities for centralized automation run. SA OS/390 publications refer to target systems or focal-point systems instead of hosts.

host (primary processor). The processor at which you enter a command (also known as the *issuing processor*)

HWMCA. Hardware Management Console Application. Application for the graphic hardware management console that monitors and controls a central processor complex. It is attached to a target processor (a system 390 microprocessor cluster) as a dedicated system console. This microprocessor uses OCF to process commands.

I

images. A grouping of processors and I/O devices that you define. You can define a single-image mode which allows a multiprocessor system to function as one central processor image.

IMS/VS. Information Management System/Virtual Storage.

inbound. In SA OS/390, messages sent to the focal-point system from the PS/2 computer or target system.

inbound gateway operator. The automation operator that receives incoming messages, commands, and responses from the outbound gateway operator at the sending system. The inbound gateway operator handles communications with other systems using a gateway session.

Information Management System/Virtual Storage (IMS/VS). A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with IMS.

INGEIO PROC. The I/O operations default procedure name; part of the SYS1.PROCLIB

initial program load (IPL). (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

initialize automation. SA OS/390-provided automation that issues the correct OS/390 start command for each subsystem when SA OS/390 is initialized. The automation ensures that subsystems are started in the order specified in the automation control file and that prerequisite applications are functional.

input/output support processor (IOSP). The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.

Interactive System Productivity Facility (ISPF). An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user.

interested operator list. The list of operators who are to receive messages from a specific target system.

internal token. A *logical token* (LTOK); name by which the I/O resource or object is known; stored in IODF.

IOCDS. I/O configuration data set. The data set that describes the I/O configuration.

I/O Ops. See *I/O operations*

IOSP. Input/Output Support Processor.

I/O operations. The part of SA OS/390 that provides you with a single point of logical control for managing connectivity in your active I/O configurations. I/O operations takes an active role in detecting unusual conditions and lets you view and change paths between a processor and an I/O device, using dynamic switching (the ESCON director).

I/O resource number. Combination of channel path identifier (CHPID), device number, etc. See internal token.

IPL. Initial program load.

ISA. Industry Standard Architecture.

ISPF. Interactive System Productivity Facility.

ISPF console. From this 3270-type console you are logged onto ISPF to use the runtime panels for I/O operations and SA OS/390 customization panels.

issuing host. See *primary host*; the base program at which you enter a command for processing.

J

JCL. Job control language.

JES. Job entry subsystem.

job. (1) A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. (2) An address space.

job control language (JCL). A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES). A facility for spooling, job queuing, and managing I/O. In SA OS/390 publications, JES refers to JES2 or JES3, unless distinguished as being either one or the other.

K

Kanji. An ideographic character set used in Japanese. See also *double-byte character set*

L

LAN. Local area network.

line mode. A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with *full-screen mode*.

link. (1) In SNA, the combination of the link connection and the link stations joining network nodes; for example, a System/370 channel and its associated protocols, a serial-by-bit connection under the control of synchronous data link control (SDLC). (2) In SA OS/390, link connection is the physical medium of transmission.

link-attached. Describes devices that are physically connected by a telecommunication line. Contrast with *channel-attached*.

local. Pertaining to a device accessed directly without use of a telecommunication line. Synonymous with *channel-attached*.

local area network (LAN). (1) A network in which a set of devices is connected for communication. They can be connected to a larger network. See also *token ring*. (2) A network in which communications are limited to a moderately sized geographic area such as a single office building, warehouse, or campus, and which do not generally extend across public rights-of-way.

logical partition (LP). A subset of the processor hardware that is defined to support an operating system. See also *logically partitioned (() mode*.

logical switch number (LSN). Assigned with the switch parameter of the CHPID macro of the IOCP

logical token (LTOK). Resource number of an object in the IODF

logical unit (LU). In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions — one with an SSCP and one with another LU — and may be capable of supporting many sessions with other LUs. See also *physical unit (PU)* and *system services control point (SSCP)*.

logical unit (LU) 6.2. A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient use of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation. Synonym for advanced program-to-program communications (APPC).

logically partitioned (LPAR) mode. A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with *basic mode*.

LOGR. The sysplex logger.

LP. Logical partition.

LPAR. Logically partitioned (mode).

LU. Logical unit.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU 6.2. Logical unit 6.2.

LU 6.2 session. A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro.

M

MCA. Micro Channel* architecture.

MCS. Multiple console support.

member. A specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

NetView message automation table. A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA OS/390 entries in the NetView automation table trigger an SA OS/390 response to target system conditions.

message class. A number that SA OS/390 associates with a message to control routing of the message. During automated operations, the classes associated with each message issued by SA OS/390 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 OS/390 process of sending messages generated at an SA OS/390 target system to the SA OS/390 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). An OS/390 table that screens all messages sent to the OS/390 console. The MPF compares these messages with a customer-defined list of messages on which to automate, suppress from the OS/390 console display, or both, and marks messages to automate or suppress. Messages are then broadcast on the subsystem interface (SSI).

message suppression. The ability to restrict the amount of message traffic displayed on the OS/390 console.

Micro Channel architecture. The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

microprocessor. A processor implemented on one or a small number of chips.

migration. Installation of a new version or release of a program to replace an earlier version or release.

MP. Multiprocessor.

MPF. Message processing facility.

Multiple Virtual Storage (MVS). An IBM licensed program. MVS, which is the predecessor of OS/390, is an operating system that controls the running of programs on a System/390 or System/370 processor. MVS includes an appropriate level of the Data Facility Product (DFP) and Multiple Virtual Storage/Enterprise Systems Architecture System Product Version 5 (MVS/ESA SP5)

multiprocessor (MP). A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application. An application program that has various functions distributed across OS/390 images in a multisystem environment.

multisystem environment. An environment in which two or more OS/390 images reside in one or more processors, and programs on one image can communication with programs on the other images.

MVS. Multiple Virtual Storage, predecessor of OS/390.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

MVS/JES2. Multiple Virtual Storage/Job Entry System 2. An OS/390 subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

MVS/ESA SP. Multiple Virtual Storage/Enterprise Systems Architecture System Product.

MVS/XA. Multiple Virtual Storage for Extended Architecture.

N

NAU. (1) Network accessible unit. (2) Network addressable unit.

NCCF. Network Communications Control Facility.

NCP. (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with *ACF/NCP*. (2) Network control program (general term).

NetView. An IBM licensed program used to monitor a network, manage it, and diagnose network problems. NetView consists of a command facility that includes a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the session monitor, hardware monitor, and terminal access facility (TAF) network management applications are built.

network accessible unit (NAU). A logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with *network addressable unit*.

network addressable unit (NAU). Synonym for *network accessible unit*.

NetView automation procedures. A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program.

NetView Command list language. An interpretive language unique to NetView that is used to write command lists.

NetView (NCCF) console. A 3270-type console for NetView commands and runtime panels for system operations and processor operations

NetView hardware monitor. The component of NetView that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. Formerly called *network problem determination application*

NetView log. The log in which NetView records events pertaining to NetView and SA OS/390 activities.

NetView-NetView task (NNT). The task under which a cross-domain NetView operator session runs. Each NetView program must have a NetView-NetView task to establish one NNT session. See *operator station task*

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.

NetView-NetView task session. A session between two NetView programs that runs under an NetView-NetView task. In SA OS/390, NetView-NetView task sessions are used for communication between focal point and remote systems.

network. (1) An interconnected group of nodes. (2) In data processing, a user application network. See *SNA network*.

Network Communications Control Facility (NCCF). The operations control facility for the network. NCCF consists of a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the network management applications NLDM and NPDA are built. NCCF is a precursor to the NetView command facility.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

Networking NetView. In SA OS/390 the NetView that performs network management functions, such as managing the configuration of a network. In SA OS/390 it is common to also route alerts to the Networking NetView.

Network Problem Determination Application (NPDA). An NCCF application that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display methods. The alert manager for the network. The precursor of the NetView hardware monitor.

NMC focal point system. See *focal point system*

NMC workstation. The NMC workstation is the primary way to dynamically monitor SA OS/390 systems. From the windows, you see messages, monitor status, view trends, and react to changes before they cause problems for end users. You can use multiple windows to monitor multiple views of the system.

NIP. Nucleus initialization program.

NNT. NetView-NetView task.

notification message. An SA OS/390 message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialog.

notification operator. A NetView console operator who is authorized to receive SA OS/390 notification messages. Authorization is made through the customization dialog.

NPDA. Network Problem Determination Application.

NPDA focal point system. See *focal point system*

NTRI. NCP/token-ring interconnection.

nucleus initialization program (NIP). The program that initializes the resident control program; it allows the operator to request last-minute changes to certain options specified during system generation.

O

OCA. In SA OS/390, operator console A, the active operator console for a target system. Contrast with *OCB*.

OCB. In SA OS/390, operator console B, the backup operator console for a target system. Contrast with *OCA*.

OCF. Operations command facility.

OCF-based processor. A central processor complex that uses an operations command facility for interacting with human operators or external programs to perform operations management functions on the CPC. Contrast with *screen-oriented processor*.

OPC/A. Operations Planning and Control/Advanced.

OPC/ESA. Operations Planning and Control/Enterprise Systems Architecture.

operating system (OS). Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

operations. The real-time control of a hardware device or software function.

operations command facility (OCF). A facility of the central processor complex that accepts and processes operations management commands.

Operations Planning and Control/Advanced (OPC/A). A set of IBM licensed programs that automate, plan, and control batch workload. OPC/A analyzes system and workload status and submits jobs accordingly.

Operations Planning and Control/ESA (OPC/ESA). A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

operator. (1) A person who keeps a system running. (2) A person or program responsible for managing activities controlled by a given piece of software such as OS/390, the NetView program, or IMS. (3) A person who operates a device. (4) In a language statement, the lexical entity that indicates the action to be performed on operands.

operator console. (1) A functional unit containing devices that are used for communications between a computer operator and a computer. (T) (2) A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and I/O operations and to monitor system operation. (3) In SA OS/390, a console that displays output from and sends input to the operating system (OS/390, VM, VSE, or LINUX). Also called *operating system console*. In the SA OS/390 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.

OS/390 component. A part of OS/390 that performs a specific OS/390 function. In SA OS/390, component refers to entities that are managed by SA OS/390 automation.

OS/390 subsystem. Software products that augment the OS/390 operating system. JES and TSO/E are examples of OS/390 subsystems. SA OS/390 includes automation for some OS/390 subsystems.

OS/390 system. An OS/390 image together with its associated hardware, which collectively are often referred to simply as a system, or OS/390 system.

OSA. I/O operations can display the open system adapter (OSA) channel logical definition, physical attachment, and status. You can configure an OSA channel on or off.

OST. Operator station task.

outbound. In SA OS/390, messages or commands from the focal-point system to the PS/2 computer or target system.

outbound gateway operator. The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

P

page. (1) The portion of a panel that is shown on a display surface at one time. (2) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel. (1) A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics. (2) By using SA OS/390 panels you can see status, type commands on a command line using a keyboard, configure your system, and passthru to other consoles. See also *help panel*. (3) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with *screen*.

parallel channels. Parallel channels operate in either byte (BY) or block (BL) mode. You can change connectivity to a parallel channel operating in block mode.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

partition. (1) A fixed-size division of storage. (2) In VSE, a division of the virtual address area that is available for program processing. (3) On an IBM Personal Computer fixed disk, one of four possible storage areas of variable size; one can be accessed by DOS, and each of the others may be assigned to another operating system.

partitionable CPC. A CPC that can be divided into 2 independent CPCs. See also *physical partition, single-image mode, MP, side*.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called *members*, each of which can contain a program, part of a program, or data.

passive monitoring. In SA OS/390, the receiving of unsolicited messages from OS/390 systems and their resources. These messages can prompt updates to resource status displays. See also *active monitoring*

path. Communication link (either NetView or PS/2) between a processor and console

PCE. Processor controller. Also known as the “support processor” or “service processor” in some processor families.

PDS. Partitioned data set.

physical partition. Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physical unit (PU). In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

physically partitioned (PP) configuration. A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, water, and maintenance boundaries. Contrast with *single-image (SI) configuration*.

POI. Program operator interface.

policy. The automation and monitoring specifications for an SA OS/390 enterprise. See *SA OS/390 policy*

policy database. The database where the automation policy is recorded.

POR. Power-on reset.

port. (1) System hardware to which the I/O devices are attached. (2) On an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you can use commands to change the switch configuration. (3) An access point (for example, a logical unit) for data entry or exit. (4) A functional unit of a node through which data can enter or leave a data network. (5) In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices. (6) power-on reset (POR) (7) A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system.

PP. Physically partitioned (configuration).

PPT. Primary POI task.

primary host. The base program at which you enter a command for processing.

primary POI task (PPT). The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

primary system. A system is a primary system for an application if the application is normally meant to be running there. SA OS/390 starts the application on all the primary systems defined for it.

problem determination. The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environment failure such as a power loss, or user error.

processor controller. Hardware that provides support and diagnostic functions for the central processors.

processor operations. The part of SA OS/390 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.

processor operations control file. Named by your system programmer, this file contains configuration and customization information. The programmer records the name of this control file in the processor operations file generation panel ISQDPG01.

processor operations workstation. This 3270-type display echoes commands run by automation and relays commands from the focal point system to the target processor, and messages from the target processor to the focal point system. The display can also be used as operator consoles. The workstation attaches to a processor through the processor controller.

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 *(*.

ProcOps. See *Processor Operations*

product automation. Automation integrated into the base of SA OS/390 for the products DB2, CICS, IMS, OPC (formerly called *features*.

program to program interface (PPI). A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

protocol. In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

proxy resource. A resource defined like an entry type *Application* representing a processor operations target system.

PR/SM. Processor Resource/Systems Manager.

PU. Physical unit.

R

remote system. A system that receives resource status information from an SA OS/390 focal point system. An SA OS/390 remote system is defined as part of the same SA OS/390 enterprise as the SA OS/390 focal point system to which it is related.

requester. A requester is a workstation software, which enables users to log on to a domain, that is, to the server(s) belonging to this domain, and use the resources in this domain. After the log on to a domain, users can access the shared resources and use the processing capability of the server(s). Because the bigger part of shared resources is on the server(s), users can reduce hardware investment.

resource. (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In NetView, any hardware or software that provides function to the network. (3) In SA OS/390, any OS/390 application, OS/390 component, job, device, or target system capable of being monitored or automated through SA OS/390.

Resource Access Control Facility (RACF). A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.

resource group. A physically partitionable portion of a processor. Also known as a *side*.

Resource Monitoring Facility (RMF). A program that measures and reports on the availability and activity of system hardware and software resources, such as processors, devices, storage, and address spaces. RMF can issue reports about system performance problems as they occur.

Resource Object Data Manager (RODM). A data cache manager designed to support process control and automation applications. RODM provides an in-memory data cache for maintaining real-time data in an address space that is accessible by multiple applications. RODM also allows an application to query an object and receive a rapid response and act on it.

resource token. A unique internal identifier of an ESCON resource or resource number of the object in the IODF

restart automation. SA OS/390-provided automation that monitors subsystems to ensure that they are running. If a subsystem fails, SA OS/390 attempts to restart it according to the policy in the automation control file.

Restructured Extended Executor (REXX). An interpretive language used to write command lists.

return code. A code returned from a program used to influence the issuing of subsequent instructions.

REXX. Restructured Extended Executor.

REXX procedure. A command list written with the Restructured Extended Executor (REXX) which is an interpretive language.

RMF. Resource Measurement Facility.

RODM. Resource Object Data Manager.

S

SAF. Security Authorization Facility.

SA OS/390 automation focal point system. See *focal point system*

SA OS/390 customization dialog. An ISPF application through which the SA OS/390 policy administrator defines policy for individual OS/390 systems and builds automation control data.

SA OS/390 customization focal point system. See *focal point system*

SA OS/390 enterprise. The group of systems and resources defined in the customization dialog under one enterprise name. An SA OS/390 enterprise consists of connected OS/390 systems running SA OS/390.

SA OS/390 focal point system. See *focal point system*.

SA OS/390 policy. The description of the systems and resources that make up an SA OS/390 enterprise, together with their monitoring and automation definitions.

SA OS/390 policy administrator. The member of the operations staff who is responsible for defining SA OS/390 policy.

SA OS/390 satellite. If you are running two NetViews on an OS/390 system to split the automation and networking functions of NetView, it is common to route alerts to the Networking NetView. For SA OS/390 to process alerts properly on the Networking NetView, you must install a subset of SA OS/390 code, called an *SA OS/390 satellite* on the Networking NetView.

SCA. In SA OS/390, system console A, the active system console for a target hardware. Contrast with *SCB*.

SCB. In SA OS/390, system console B, the backup system console for a target hardware. Contrast with *SCA*.

screen. Deprecated term for display panel.

screen handler. In SA OS/390, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

SDF. Status display facility.

SDLC. Synchronous data link control.

SDSF. System Display and Search Facility.

secondary system. A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable.
SA OS/390 does not start the application on its secondary systems.

server. A server is a workstation, that shares resources, which include directories, printers, serial devices, and computing powers.

service period. Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

service threshold. An SA OS/390 policy setting that determines when to notify the operator of deteriorating service for a resource. See *alert threshold* and *warning threshold*

service language command (SLC). The line-oriented command language of processor controllers or service processors.

service processor (SVP). The name given to a processor controller on smaller System/370 processors.

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 OS/390-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

side. A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

Simple Network Management Protocol (SNMP) . An IP based industry standard protocol to monitor and control resources in an IP network.

single image. A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.

single-image (SI) mode. A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with *physically partitioned (PP) configuration*.

SLC. Service language command.

SMP/E. System Modification Program Extended.

SNA. Systems Network Architecture.

SNA network. In SNA, the part of a user-application network that conforms to the formats and protocols of systems network architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNMP. Simple Network Management Protocol (a TCP/IP protocol). A protocol that allows network management by elements, such as gateways, routers, and hosts. This protocol provides a means of communication between network elements regarding network resources.

solicited message. An SA OS/390 message that directly responds to a command. Contrast with *unsolicited message*.

SRPI. Server-Requester Programming Interface.

SSCP. System services control point.

SSI. Subsystem interface.

start automation. SA OS/390-provided automation that manages and completes the startup process for subsystems. During this process, SA OS/390 replies to prompts for additional information, ensures that the startup process completes within specified time limits, notifies the operator of problems, if necessary, and brings subsystems to an UP (or ready) state.

startup. The point in time at which a subsystem or application is started.

status. The measure of the condition or availability of the resource.

status focal point system. See *focal point system*

status display facility (SDF). The system operations part of SA OS/390 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 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 OS/390 to map and manage storage on a coupling facility. See cache structure, list structure, and lock structure.

subgroup. A named set of systems. A subgroup is part of an SA OS/390 enterprise definition and is used for monitoring purposes.

SubGroup entry. A construct, created with the customization dialog, used to represent and contain policy for a subgroup.

subsystem. (1) A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system. (2) In SA OS/390, an OS/390 application or subsystem defined to SA OS/390.

subsystem interface. The OS/390 interface over which all messages sent to the OS/390 console are broadcast.

support element. A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

support processor. Another name given to a processor controller on smaller System/370 processors; see *service processor*.

SVP. Service processor.

switches. ESCON directors are electronic units with ports that dynamically switch to route data to I/O devices. The switches are controlled by I/O operations commands that you enter on a workstation.

switch identifier. The switch device number (swchdevn), the logical switch number (LSN) and the switch name

symbolic destination name (SDN). Used locally at the workstation to relate 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 non-switched links. The configuration of the link connection may be point-to-point, multi-point, 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.

SysOps. See *System Operations*

sysplex. A set of OS/390 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, OS/390 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 which 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 OS/390 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 OS/390 generic name for the IBM Sysplex Timer (9037).

system. In SA OS/390, system means a focal point system (OS/390) or a target system (MVS, VM, VSE, TPF or CF).

System Automation for OS/390. The full name for SA OS/390.

system console. (1) A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system. (2)

A logical device used for the operation and control of hardware functions (for example, IPL, alter/display, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor. (3) In SA OS/390, the hardware system console for processor controllers or service processors of processors connected using SA OS/390. In the SA OS/390 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 OS/390 you can select SDSF from a pull-down menu to see the resources' status, view the OS/390 system log, see WTOR messages, and see active jobs on the system.

System entry type. A construct, created with the customization dialog, 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 OS/390 system.

system operations. The part of SA OS/390 that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC.

system services control point (SSCP). In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

System/390 microprocessor cluster. A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

T

TAF. Terminal access facility.

target. A processor or system monitored and controlled by a focal point system.

target control task. In SA OS/390, 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 OS/390, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with *target system*.

target system. (1) In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system. (2) In SA OS/390, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

task. (1) A basic unit of work to be accomplished by a computer. (2) In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA OS/390 tasks are NetView tasks. See also *communications task*, *message monitor task*, and *target control task*.

telecommunication line. Any physical medium, such as a wire or microwave beam, that is used to transmit data.

terminal access facility (TAF). (1) A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA OS/390 customization panels so you don't have to set them up each time you want to use them. (2) In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

terminal emulation. The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

threshold. A value that determines the point at which SA OS/390 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 OS/390, MVS/ESA, and MVS/XA.

Time-Sharing Option/Extended (TSO/E). An option of OS/390 that provides conversational time-sharing from remote terminals. TSO/E allows a wide variety of users to perform many different kinds of tasks. It can handle short-running applications that use fewer sources as well as long-running applications that require large amounts of resources.

timers. A NetView command that issues a command or command processor (list of commands) at a specified time or time interval.

TOD. Time of day.

token ring. A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network product.

TP. Transaction program.

transaction program. In the VTAM program, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the applications program using protocols defined by that application program. The application program, in turn, could request services from the VTAM program by issuing the APPCCMD macro instruction.

transitional automation. the actions involved in starting and stopping subsystems and applications which have been defined to SA OS/390. 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. On top of requests, triggers, in combination with events, are used to control the starting and stopping of applications in a single system or a Parallel Sysplex. Triggers act as inhibitors for the requested action.

TSO. Time Sharing Option.

TSO console. From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for I/O operations and SA OS/390 customization panels.

TSO/E. TSO Extensions.

U

UCB. The unit control block; an MVS/ESA data area that represents a device and which is used for allocating devices and controlling I/O operations.

unsolicited message. An SA OS/390 message that is not a direct response to a command. Contrast with *solicited message*.

user task. An application of the NetView program defined in a NetView TASK definition statement.

V

Virtual Machine/System Product (VM/SP). An IBM licensed program. It is an operating system that manages the resources of a real processor to provide virtual machines to end users. As a time-sharing system control program, it consists of the virtual machine control program (CP), the conversational monitor system (CMS), the group control system (GCS), and the interactive problem control system (IPCS).

Virtual Storage Extended (VSE). An IBM licensed program whose full name is Virtual Storage Extended/Advanced Function. It is an operating system that controls the execution of programs.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with *ACF/VTAM*.

VM/ESA*. Virtual Machine/Enterprise Systems Architecture*

VM/SP. Virtual Machine/System Product.

VM/SP HPO. Virtual Machine/System Product High Performance Option.

VSE. Virtual Storage Extended.

VTAM. Virtual Telecommunications Access Method.

W

warning threshold. An application service value that determines the level at which SA OS/390 changes the associated icon on the NMC workstation to the warning color. See *alert threshold*

workstation. In SA OS/390 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 OS/390 operator console. This request is made by an application and is handled by the WTO processor, which is part of the OS/390 supervisor program.

write-to-operator-with-reply (WTOR). A request to send a message to an operator at the OS/390 operator console which 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 OS/390 supervisor program.

WTO. Write-to-Operator.

WTOR. Write-to-Operator-with-Reply.

WWV. The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

X

XCF. Cross-system coupling facility.

XCF couple data set. The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See *sysplex couple data set*

XCF group. A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

XRF. Extended recovery facility.

Numerics

3270 connection card. A 3270 emulator card installed in a workstation used to connect the workstation to the target system. Four 3270 connection cards can be installed in a workstation.

37x5. A 3705, 3725, or 3745 communication controller.

3x74. A 3174 or 3274 subsystem control unit. A control unit to which 3270-family display stations are attached.

Index

Readers' Comments — We'd Like to Hear from You

System Automation for OS/390
Planning and Installation
Version 2 Release 2

Publication No. SC33-7038-05

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

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.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You

SC33-7038-05

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

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

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Program Number: 5645-006

Printed in U.S.A.

SC33-7038-05

Spine information:

IBM System Automation for OS/390 **Planning and Installation** Version 2 Release 2 SC33-7038-05