z/OS and OS/390

Managed System Infrastructure for Operations
Setting Up and Using

SC33-7968-06

z/OS and OS/390

Managed System Infrastructure for Operations
Setting Up and Using

SC33-7968-06

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 369.

Sixth Edition (October 2004)

This edition applies to Version 1 Release 1 of z/OS (Program Number 5694-A01) and to Version 2 Release 10 of OS/390 (Program Number 5647–A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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 15. Debugging and support commands 189

Part 5. Setup reference 201

Chapter 16. msys for Operations definition statements reference. 203

Part 6. Automation-related messages 217

Chapter 17. Messages. 219

Part 7. Appendixes 227

Appendix A. Making security definitions using the command authorization table 229

Appendix B. Return codes. 255

Appendix C. Coexistence of msys for Operations and SA OS/390 releases . 265

Appendix D. msys for Operations customization checklist 267

Figures

Tables

About this book

This book describes Managed System Infrastructure for Operations (msys for Operations). msys for Operations provides a simplified operating environment for Parallel Sysplexes. This simplification relates to two areas: automation of recovery actions, and a command interface that allows you to perform complex tasks using function keys.

Throughout this book, unless a particular release is named, any references to z/OS also apply to OS/390, and *vice versa*. In these cases, this means that references to the two operating systems are interchangeable.

| **Note:** The sections marked with a revision bar '|' describe enhanced automation
| functions shipped with SPE OW50146 (03/2002). This functionality is only
| available if you have installed OW50146.

| **Note:** Those sections marked with the following revision bar '1' describe enhanced
| automation functions shipped with SPE OW56107 (05/2003). This
| functionality is only available if you have installed OW56107.

+ **Note:** Those sections marked with the following revision bar '+' describe enhanced
+ automation functions shipped with SPE OA08154 (10/2004). This
+ functionality is only available if you have installed OA08154.

Who should use this book

This book is intended for sysplex operators and system programmers. Sysplex operators will use the enhanced command interface of msys for Operations for their daily work; system programmers are primarily responsible for installation and customization of the product, but will also make use of the command interface.

Both types of users are expected to have a basic knowledge of Parallel Sysplexes.

How this book is organized

This book contains the following parts:

- Part 1, "Introducing Managed System Infrastructure for Operations," on page 1 describes the general concept and the functional scope of msys for Operations, including brief explanations of the relevant sysplex terms and concepts. It is intended as a general introduction for both operators and system programmers. System programmers can also use it to decide which automatic recovery actions are to be enabled.
- Part 2, "Setting up msys for Operations," on page 27 describes the installation and customization of msys for Operations. It also describes how to protect system resources from unauthorized use. This part is of interest mainly to system programmers.
- Part 3, "Using the msys for Operations operator interface," on page 101 describes the msys for Operations operator interface.
- Part 4, "Command reference," on page 119 contains descriptions of all the commands that are available with msys for Operations. This part is divided into three chapters:

  - Chapter 13, “General commands,” on page 121 contains the operator interface commands.
  - Chapter 14, “Sysplex-related commands,” on page 133 describes the commands that are used for sysplex management.
  - Chapter 15, “Debugging and support commands,” on page 189 describes commands that are used for problem determination. These commands will mainly be used by system programmers.
- Part 5, “Setup reference,” on page 201 contains information for the system programmer about additional security options and the configuration statements of NetView System Services (NVSS).

Where to find more information

The following table lists publications that are related to z/OS:

Title	Order Number
z/OS MVS Setting Up a Sysplex	SA22-7625
z/OS MVS Programming: Sysplex Services Guide	SA22-7617

The following table lists publications that are related to OS/390:

Title	Order Number
OS/390 MVS™ Setting Up a Sysplex	GC28-1779
OS/390 MVS Programming: Sysplex Services Guide	GC28-1771

Throughout this book, these publications will be referred to as *MVS Setting Up a Sysplex* and *MVS Programming: Sysplex Services Guide*. You should refer to the publication that is related to the operating system you are using, that is, either z/OS or OS/390.

z/OS msys for Operations Home page

For the latest news on msys for Operations, visit the msys for Operations home page at http://www.ibm.com/servers/eserver/zseries/msys/msysops

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]

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.

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

Part 1. Introducing Managed System Infrastructure for Operations

This part describes the goals and scope of the functions that are offered by Managed System Infrastructure for Operations (msys for Operations). It is intended for both system programmers and operators. It has the following chapters:

- Chapter 1, “General concept,” on page 3
- Chapter 2, “Sysplex functions,” on page 9

Chapter 1. General concept

msys for Operations provides a simplified environment for operating Parallel Sysplex®es. Managing a Parallel Sysplex, although easier than administering the member systems independently, is still a rather complex task. Recovery and maintenance actions may require the operator to enter long chains of complicated commands. These chains may branch depending on the status of the resources to be managed. Even gathering the necessary information can be difficult.

Consider the following examples:

1. After switching from a primary to an alternate couple data set (CDS), the system issues the message IXC263I, which indicates that an alternate CDS is no longer available. Because this would entail a single point of failure, the operator must provide a new alternate CDS. To allocate and format this new alternate CDS, the operator must call the formatting tool for CDSs. In this call several parameters must be specified, and it is essential that the parameter values for the new alternate CDS are not smaller than those for the old alternate CDS (which is now the primary). Next, the new alternate CDS must be introduced to the system with the SETXCF COUPLE,ACOUPLE command.
2. To remove a coupling facility (CF) from the sysplex, the sender paths of all systems to that CF must be set OFFLINE. Currently the operator must do this separately for every system. There is a risk that the operator may erroneously disable the sender paths of the backup CF that maintains operations during the removal of the other CF. If this happens, the sysplex will go down.

msys for Operations reduces the complexities of operating a sysplex as follows:

- A number of recovery actions can be completely automated. These include:
  - Creating or re-creating missing alternate couple data sets

    msys for Operations reacts to message IXC263I, calls the formatting tool with the parameters of the current primary CDS, and introduces the new alternate CDS to the system. No operator intervention is required.
  - Expanding the couple data sets for the system logger in case of a directory shortage
  - Resolving write-to-operator (WTO) and write-to-operator-with-reply (WTOR) buffer shortages
  - System log recovery
  - Handling long-running ENQs
  - Hung command recovery

  Each automatic recovery action can be enabled or disabled separately. By default, all actions are disabled. A number of them can be customized. Enabling, disabling, and customizing is performed by editing a special data set member that is described in detail in “Editing the customization member AOFCUST” on page 46.
- msys for Operations provides three global commands (INGPLEX, INGCF, and INGHC) that:
  - Display comprehensive information about sysplex components in a structured way (for example, couple data sets, policies, coupling facilities, structures, members of the sysplex)

- Allow you to use function keys to perform complex command sequences that are based on the information displayed (for example, draining a coupling facility, or reintegrating a coupling facility into the sysplex)

  One step in the draining process is to set the sender paths of the respective CF offline, as described in Example 2 on page 3. With msys for Operations, this can be done for all members of the sysplex with one key stroke. There is also no risk of confusing the CF to be removed with the backup CF.
- Enable you to use the IBM® Health Checker for z/OS and Sysplex (HealthChecker). This is a tool that checks the current, active z/OS and sysplex settings and definitions for an image, and compares their values to those either suggested by IBM or defined by you.

The INGPLEX and INGCF commands support actions that have an impact on the sysplex configuration. You can control access to these actions through a security product such as Resource Access Control Facility (RACF®). For details, see "Command authorization" on page 77.

For a short explanation of the relevant sysplex concepts and a more detailed description of the functional scope of msys for Operations, see Chapter 2, "Sysplex functions," on page 9.

The structure of msys for Operations and its position within the sysplex

This section briefly sketches the internal structure of msys for Operations and its position within the sysplex.

Internal structure of msys for Operations

msys for Operations combines the functions of two licensed products, NetView® for OS/390 and System Automation for OS/390 (SA OS/390). The SA OS/390 part of msys for Operations provides the sysplex specific recovery functions and commands. In this book, it will be referred to as *Sysplex Functions*. It runs on top of the NetView part that supplies the infrastructure with basic services such as the command interface, triggering of automated recovery, and logging. It will be referred to as *NVSS* (NetView System Services) in this book. For more information on NVSS, see "Task and message automation" on page 5.

Because msys for Operations contains a subset of the functions offered by the licensed products, you can migrate to them from it. The relation between the licensed products is similar to that between the two parts of msys for Operations. A licensed NetView is a prerequisite for the automation functions of the licensed SA OS/390. You should note that the formation of msys for Operations from these two licensed products is still visible in its panels and help texts. Therefore, you may encounter 'NetView' or 'SA OS/390' where you would expect 'msys for Operations'.

msys for Operations within the sysplex

Figure 1 on page 5 shows a sample sysplex with msys for Operations installed:

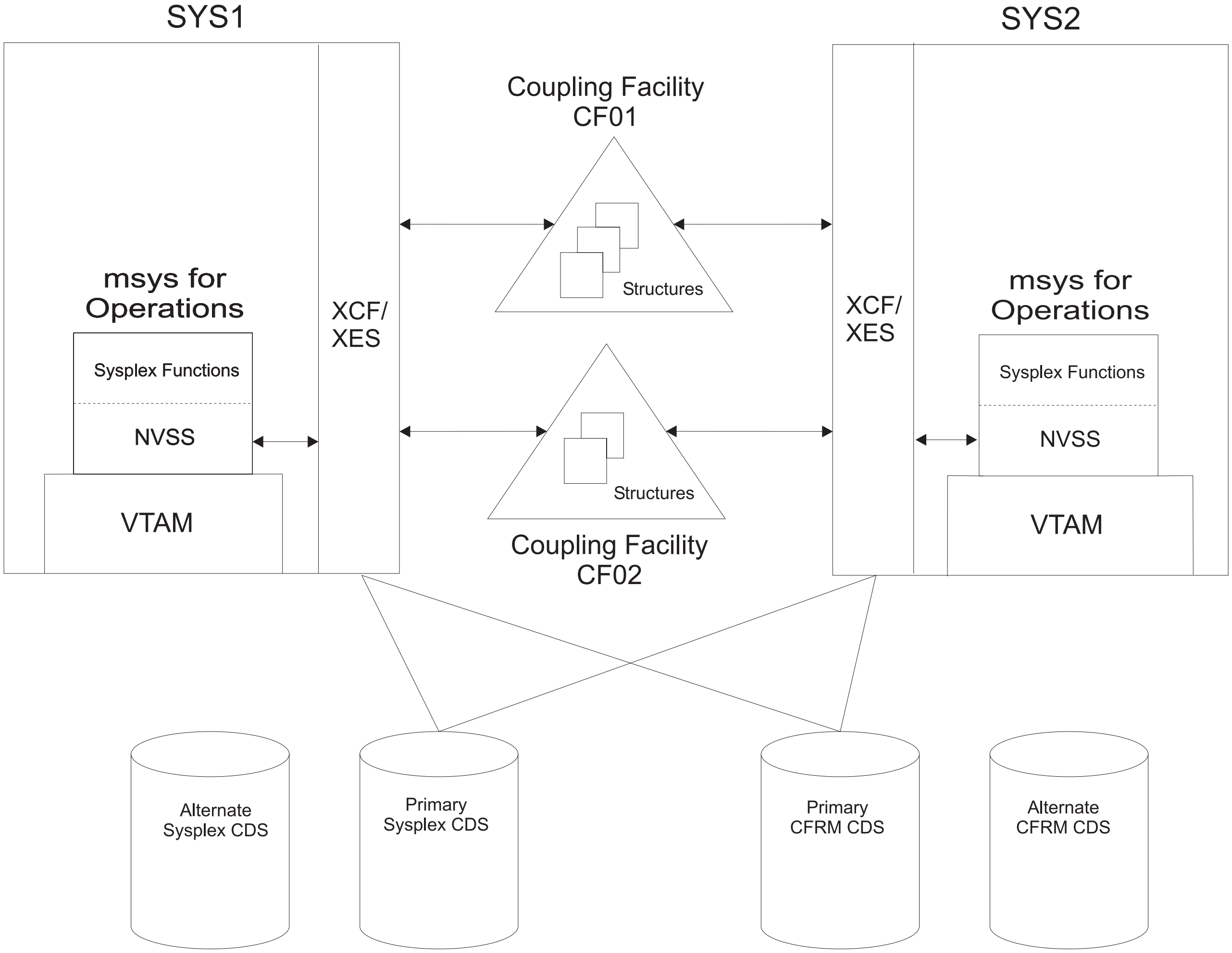


Figure 1. msys for Operations in a simple Parallel Sysplex configuration

This Parallel Sysplex consists of two Multiple Virtual Storage (MVS) images and two coupling facilities. Cross-system coupling facility (XCF) (Cross-system Coupling Facility) and XES (Cross-system Extended Services®) are the sysplex components of MVS. XCF supplies the basic sysplex services, and XES manages the coupling facilities and structures. The sysplex components of both MVS images must be able to access the primary sysplex and CFRM (Coupling Facility Resource Management) couple data sets. For more information on couple data sets and coupling facilities, see Chapter 2, “Sysplex functions,” on page 9.

msys for Operations must be installed on every system of the Parallel Sysplex. Because NVSS, and therefore msys for Operations, is a Virtual Telecommunications Access Method (VTAM®) application, msys for Operations must be defined to VTAM during the installation (see “Preparing VTAM” on page 43). The individual instances of msys for Operations use XCF and XES for mutual communication (through an XCF group) and for triggering maintenance and recovery actions.

Task and message automation

This section provides a short introduction to some of the key concepts of the NVSS part of msys for Operations.

Understanding tasks

msys for Operations is internally organized through *tasks*. If you are a system programmer, it will be helpful for you to have a basic understanding of this internal structure because you will have to specify names and attributes for certain tasks during installation.

A task is a process with certain properties that is defined to, and runs within, msys for Operations. Tasks can be started and stopped. Every request you put to msys for Operations is executed by a task.

One common type of task is an *operator station task* (OST). This starts when you have successfully logged on to msys for Operations. The OST establishes and maintains the online session with you. It receives your commands, executes them, and displays the messages that are sent in response. The OST is stopped when you log off. OSTs are defined to msys for Operations through the operator ID and a password, and a set of authorizations.

OSTs receive and execute the commands of a human operator. The automation procedures of msys for Operations also need tasks that execute commands for them. However, OSTs that were started by an operator logon are not suitable for these for two reasons. First, you cannot be sure that the respective user is really logged on, and second, an automation procedure needs no terminal to pass a command to the task that is to execute it.

To accommodate the needs of automation, msys for Operations provides another type of task that is called an *autotask*. These perform functions similar to those of OSTs, but they do not require a terminal or online user. They are usually started immediately after the start of msys for Operations and are active until it is shut down. One important application of autotasks is message automation, which is described in the following section.

Note that autotasks and OSTs do not differ in their definition. Both are defined through an ID and certain authorizations. They differ in the way in which they are started: OSTs are started with an operator logon; autotasks are started with a command. Thus, the same task can be started as an OST and an autotask. For the definition of IDs for human operators and autotasks, see “Defining operators, passwords, and logon attributes” on page 75.

Another important type of task is the automation router task. This receives the automatable MVS messages and plays a crucial role in message automation (see “Automatic recovery and message automation”). Its name must be specified during the installation of msys for Operations (see “Customizing the initialization style sheet” on page 45).

Automatic recovery and message automation

msys for Operations performs most of its automatic recovery actions in response to certain messages that are issued by the sysplex components of MVS (XCF and XES). For these messages, msys for Operations message automation supplies a hard-coded message automation table that specifies what has to be done in response to the message, and which task should execute that response.

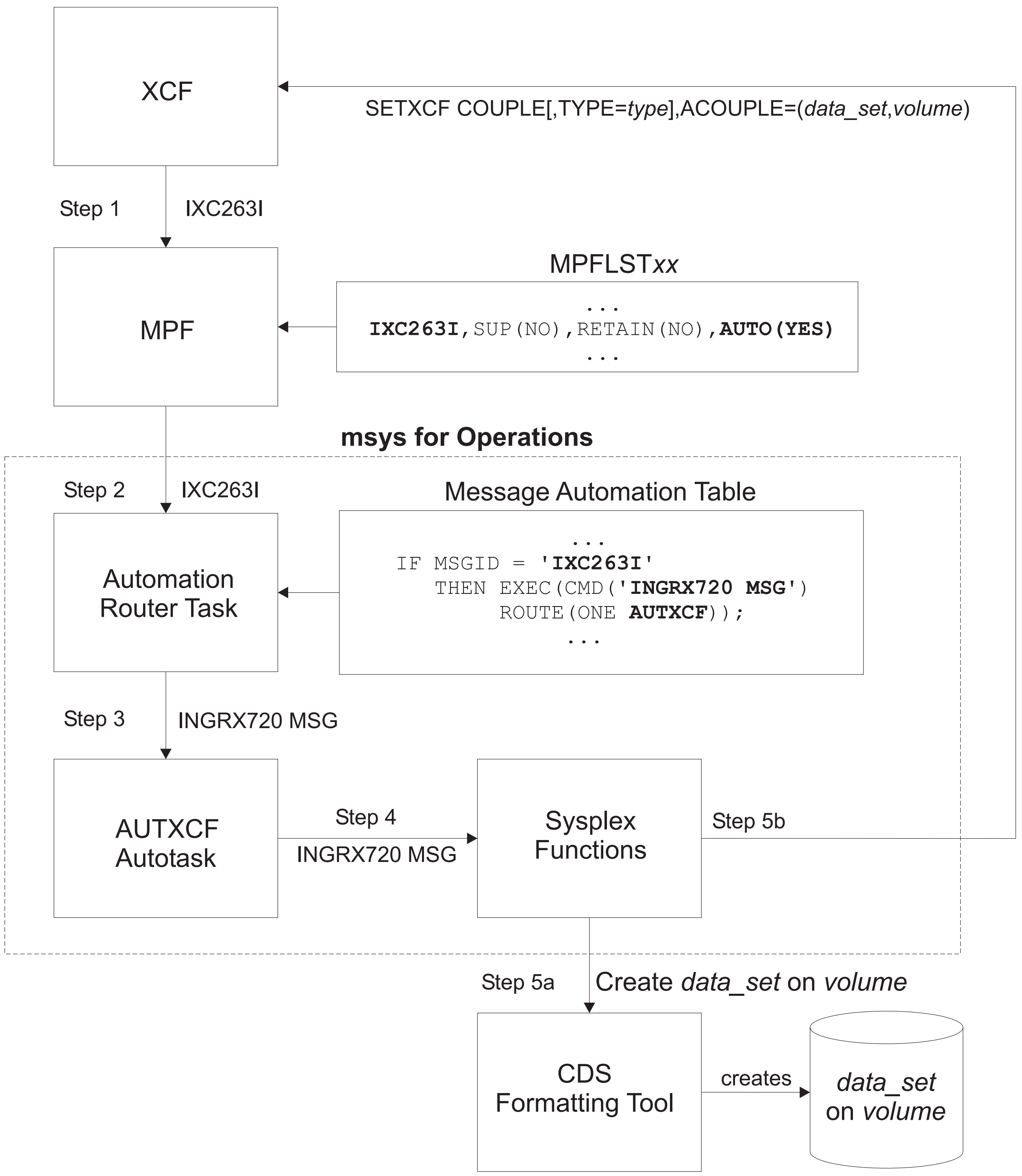
You have to define these messages as capable of being automated to the MVS message processing facility (MPF) during installation of msys for Operations. For details refer to “Updating member MPFLST*xx*” on page 38. If appropriate definitions exist and MPF is activated, it forwards the messages to msys for

Operations at run time. In msys for Operations, the automation router task receives the message, reads the automation table, and routes the command that is associated with the message to its associated task, which then issues the command.

The tasks that perform automated recovery actions are started as autotasks immediately after the start of msys for Operations, and are active as long as msys for Operations itself is running. Thus, recovery actions can be performed even when no human operator is logged on. In order to enhance performance, msys for Operations distributes the automated responses to several autotasks that work in parallel.

In example 1 on page 3, message automation works as follows (Figure 2 on page 8 illustrates the information flow):

1. After the alternate CDS for a certain CDS type has been made the primary, XCF issues the message IXC263I, which impliesthat an alternate CDS is no longer available (for details on CDSs, see “Managing couple data sets” on page 9).
2. If IXC263I has been defined as automatable, the message is forwarded to msys for Operations. The automation router task receives the message.
3. The automation router task then reads the automation table, which contains an entry for IXC263I. The command specified in that entry will create a new alternate CDS. The entry also specifies the task that is to execute that command. The command is routed to the task that is specified in the table entry.
4. The target task executes the command.
5. A new alternate CDS is created for the CDS type in question, which involves the following:
   a. Allocating the new CDS by calling the XCF formatting tool
   b. Defining the formatted data set to XCF



```
...
IXC263I,SUP(NO),RETAIN(NO),AUTO(YES)
...
```

```
   ...
IF MSGID = 'IXC263I'
   THEN EXEC(CMD('INGRX720 MSG')
        ROUTE(ONE AUTXCF));
   ...
```

Figure 2. Automation for message IXC263I

Chapter 2. Sysplex functions

This chapter gives an overview of the sysplex functions that are available with msys for Operations. These are grouped by sysplex component, for example, couple data sets or coupling facilities. Each subsection describes the functions of one group, and how they can be customized. All customization information must be defined in the member AOFCUST, which is described in detail in "Editing the customization member AOFCUST" on page 46.

At the beginning of each subsection, relevant sysplex concepts are briefly explained, with the emphasis on the actions that have an impact on the sysplex configuration. For more information about Parallel Sysplexes, see *MVS Setting Up a Sysplex* and *MVS Programming: Sysplex Services Guide*..

Display functions are only mentioned.

Managing couple data sets

Couple data sets (CDSs) contain control information about the sysplex and its resources, and are of crucial importance for the functioning of a Parallel Sysplex. Particularly important are the SYSPLEX couple data set, which contains information about the systems and the communication structure (XCF groups) of the sysplex, and the CFRM couple data set, which specifies its coupling facilities (CFs) and structures (see "Managing coupling facilities" on page 16). Every MVS system in a Parallel Sysplex must have access to these CDSs, and to those of all other implemented sysplex functions, such as SFM and Application Response Measurement (ARM).

If a member system cannot access a CDS, the corresponding sysplex function is impacted, and in some cases the sysplex will go down. It is therefore recommended that you define two CDSs to XCF for every CDS type required for the implementation of the sysplex. One of these, the *primary* CDS, is the one that is actually used. The other, which is called the *alternate* CDS, serves as a backup copy. The two CDSs contain the same data. Whenever the primary CDS changes, XCF updates the alternate CDS accordingly. If an alternate CDS is available for a certain type, XCF automatically switches to this alternate CDS whenever a member can no longer access the primary CDS.

All CDSs except the sysplex couple data set contain one or more user-defined configurations, called *policies*. For each CDS type, only one policy can be active. However, it is possible to switch the active policy at run time. Refer to "INGPLEX CDS" on page 10 for further information.

msys for Operations offers two functions for easier CDS management:

- Automated creation and recovery of alternate couple data sets for continuous availability
- INGPLEX CDS, which simplifies management of couple data sets

Ensuring continuous availability of alternate couple data sets

When an alternate CDS exists for a given CDS type and the current primary CDS fails, XCF makes this alternate the primary CDS. After this switch, however, an alternate CDS no longer exists, and if the current primary CDS also fails, the problems that were to be avoided by the creation of an alternate occur again. To avoid this single-point-of-failure situation, msys for Operations provides a recovery mechanism that tries to ensure that an alternate CDS is always available for every CDS type used.

msys for Operations creates a new alternate CDS in the following two situations:

- During initialization, msys for Operations checks that an alternate CDS is specified for every primary CDS. If there is a primary CDS for which no alternate CDS exists, msys for Operations automatically creates it.
- At run time, msys for Operations ensures that a new alternate is created whenever the current alternate has been removed or switched to the primary one.

INGPLEX CDS

INGPLEX CDS displays information about all couple data sets, including details of the respective policies, and allows you to perform the following actions for every CDS type that is required for the implementation.:

- Switch from the primary to the alternate CDS
- Define a new alternate CDS
- Change the active policy (if applicable)
- Automatically rebuild a structure after the activation of the CFRM policy

For the first two actions, INGPLEX CDS offers automatic creation of a new alternate CDS. You can also specify your own alternate CDS. For more information on INGPLEX CDS, see “CDS” on page 169.

Customization

Recovery of alternate CDSs is initiated either by the CDS function of INGPLEX or in the background (for example, at initialization time). Background recovery can be switched on and off by setting or unsetting `CDS` in the `AUTO` section of AOFCUST. Automatic re-creation with INGPLEX CDS is always enabled. It cannot be switched off in the `AUTO` section.

You must specify the spare volumes that msys for Operations may use for creating missing alternate CDSs (the `CDS` section of AOFCUST). This is also required for automatic creation with INGPLEX CDS. Every CDS type has its own pool of spare volumes. Note that if you do not define spare volumes for a CDS type, no recovery will be performed for this type. For details on the AOFCUST entries, see “Editing the customization member AOFCUST” on page 46.

You can control access to those functions of INGPLEX CDS that modify the sysplex configuration. Refer to “Command authorization” on page 77 for details.

Managing the system logger

The system logger provides a sysplex-wide logging facility. Applications that use the system logger write their log data into *log streams*. Within a Parallel Sysplex, these log streams are usually associated with a coupling facility structure. For further information about coupling facility structures, refer to “Managing coupling facilities” on page 16. By using a coupling facility log stream, members of a multisystem application can merge their logs even when residing on different systems.

When an application writes data to a log stream, this data is at first stored temporarily in the associated structure (coupling facility log stream) or a local buffer (DASD-only log stream). From there, it is off-loaded into a log stream data set that is automatically allocated by the system logger. When this log stream data set is full, the system logger allocates a second one, and so on.

The control information for the system logger, which includes a directory for the log stream data sets of every log stream, is contained in the LOGR couple data set. The total number of log stream data sets that can be allocated by the system logger is determined when the LOGR couple data set is formatted.

Two problems that can arise in connection with the log stream data sets are a shortage of directory space in the LOGR CDS and incorrect share options for the log stream data sets. msys for Operations provides the following recovery actions for these problems:

- The primary and alternate LOGR CDSs are automatically resized if there is a directory shortage
- The operator is notified if the share options for log stream data sets are not defined correctly (from z/OS 1.3 the system resolves this problem itself)

Resizing the LOGR couple data sets

The LOGR CDS contains information about the log stream data sets used by the system logger which is stored in *directory extents*. Every directory extent record can hold information on up to 168 log stream data sets. The number of directory extents available in a LOGR CDS is specified when the CDS is formatted (using the DSEXTENT parameter). When all available directory extents have been used, the system logger can no longer allocate new log stream data sets. This can cause considerable problems for applications that use the system logger.

With msys for Operations, you can avoid this situation. If you switch on logger recovery, msys for Operations automatically reformats your primary and alternate LOGR CDS with an increased DSEXTENT parameter whenever the system reports a directory shortage.

Notifying the operator of incorrect share options

Note: This section applies to z/OS 1.2 and below.
If you wish to use the system logger, you must define share options for the log stream data sets. Merging data from several systems into one coupling facility log stream requires you to specify VSAM SHAREOPTIONS(3,3) for the log stream data sets. With other share options, especially (1,3), such a merge will fail. If you manage your DASD data sets with SMS (Storage Management Subsystem), a possible cause for incorrect share options is that the data class you use for the log stream data sets is also used for other purposes that require different share options.

msys for Operations provides a control mechanism for VSAM share options. The share options are checked on a daily basis. If incorrect share options are detected, msys for Operations notifies the operator.

Customization

Automation of system logger recovery is enabled by setting LOGGER in the AUTO section of AOFCUST.

No further customization is required. Note however that if you switch on automation for system logger recovery, you also activate the recovery function for alternate CDSs (see “Ensuring continuous availability of alternate couple data sets” on page 9), even if CDS automation is switched off. For details, see “Editing the customization member AOFCUST” on page 46.

Recovery functions

Resolving a system log failure

When the system log is inactive, msys for Operations attempts to start it by issuing the WRITELOG START command, and to make it the hardcopy log by issuing the VARY SYSLOG,HARDCPY command.

SYSLOG message automation has been enhanced with a recovery function. Both functions (recovery and automation of message IEE043I) exist in parallel. Recovery takes place if the system log becomes inactive. It responds to message IEE037D following one of the messages IEE043I, IEE533E, or IEE769E, and it responds to message IEE041I.

Customization

Automation of system log recovery is enabled by setting `LOG` in the `AUTO` section of AOFCUST.

For details on the AOFCUST entries, see “Editing the customization member AOFCUST” on page 46.

Resolving WTO(R) buffer shortages

When all WTO(R) buffers are in use, it is possible that commands can no longer be processed. To resolve this, there are several options: you can extend the buffer, change the properties of the affected consoles, or cancel jobs that issue WTO(R)s.

msys for Operations provides recovery of buffer shortage in two stages. It first tries to extend the buffer and modify the console characteristics, if applicable. If this does not help, it then cancels jobs that issue WTO(R)s. You must specify which jobs can be canceled by msys for Operations if there is a buffer shortage.

Customization

Automation of buffer shortage recovery is enabled by setting `WTO` in the `AUTO` section of AOFCUST.

If you want msys for Operations to resolve a buffer shortage by cancel jobs that issue WTO(R)s in order, you must specify these jobs in the `WTOBUF` section of the AOFCUST member.

For details on the AOFCUST entries, see “Editing the customization member AOFCUST” on page 46.

Handling long-running enqueues (ENQs)

This type of recovery is divided into the following individual functions:

- Long-running enqueue recovery
- SYSIEFSD resource recovery
- ″Hung″ command recovery
- Command flooding recovery

All these recoveries can be enabled and disabled individually or globally.

The long-running enqueue recovery function lets you:

- Check which resources are blocked
- Customize automation to cancel or keep the jobs that block the resource
- Customize automation to dump the jobs before they are canceled

You can determine which resources you want to monitor. You can define a value for the maximum time a job can lock a resource while other jobs are waiting for it. If this amount of time is exceeded, recovery takes place. Identification of and elimination of these potential bottlenecks helps to reduce the risk of a Parallel Sysplex outage.

While the time definition describes an inclusion list, you also have the possibility to define an *exclusion list* of resources that are not monitored at all.

For more information about installing the ENQ function, see “ENQ section – resources to be monitored and jobs to be canceled” on page 53.

This function has been extended by three supplementary functions:

- “SYSIEFSD resource recovery”
- “″Hung″ Command Recovery”
- “Command Flooding Recovery” on page 14

SYSIEFSD resource recovery

The purpose of this function is to detect critical ENQ resources that, if held for extended periods of time, can cause commands to hang. Hung commands often result in multisystem outages. The focus of this function is on the SYSIEFSD family of resources that is involved in 98% of hung command outages:

- SYSIEFSD Q10 – this resource is required for every command. It is used to serialize changes to the CSCB chain. If any task gets this resource and then hangs, *all commands* will be locked out of the system. This also means that *all consoles* will be locked out of the system. This is because, as soon as a console issues a command after Q10 has hung, it will be waiting behind Q10, and that locks out the task that handles all MCS consoles. EMCS consoles will then also get locked out one by one as they issue a command and also get hung behind Q10. Actions taken to free up this hang cannot include issuing a command (for example, D GRS)—the task has to be terminated via CALLRTM.
- SYSIEFSD Q4 – this resource is used to serialize changes to the UCB by allocation and VARY command processing. Allocation obtains the resource as SHARED, while the VARY command obtains it exclusively. If a VARY command hangs while holding this resource, all allocations will also hang. The VARY command that is hung can be displayed and abended with the CMDS command.

If any of these resources do not execute within 10 seconds, they are considered to have hung.

″Hung″ Command Recovery

The purpose of this function is to detect hung commands that often result in multisystem outages. We distinguish two situations:

1. Commands that inhibit other commands from completing execution
2. Jobs that inhibit commands from completing execution

In either case only locked resources are taken into consideration. The recovery looks for blocked resources that have not been defined during customization. If the long-running ENQ recovery is disabled all resources, even those that have been defined during customization, are considered as not having been defined.

Because commands are executed by the master and the console address space, the recovery first looks for blockers and waiters of these address spaces. As with resources you can make similar definitions for commands (see “ENQ section – resources to be monitored and jobs to be canceled” on page 53).

In the second case the recovery does not take place immediately. Only after the threshold—the invocation after next—has been reached is the recovery action performed.

In both cases the action is identical to the long-running ENQ recovery action.

Command Flooding Recovery

The purpose of this function is to detect jobs that flood a command class. Command flooding can cause log buffer shortages and inhibits other commands from executing. Both can lead to a multisystem outage.

When all (50) TCBs that are reserved for command processing are in use, new commands are queued to the waiting queue. In this case the system issues message IEE806A which triggers this function to evaluate what jobs are causing the situation.

Jobs that just issue a set of commands, such as 200 (or more) ″VARY dev,ONLINE″ commands should *not* be considered during the evaluation. This is achieved by comparing the current and the previous snapshot of the affected command class.

Snapshot processing is scheduled when message IEE806A is trapped. The interval time between the snapshots is 3 seconds by default (see “ENQ section – resources to be monitored and jobs to be canceled” on page 53 for details about adjusting this value if necessary). The interval should give these jobs enough time to finish issuing commands before the first snapshot is taken. Only jobs that issue commands on two consecutive snapshots become subject of the recovery action.

Before the recovery action takes place, the number of commands that are issued by the job must exceed a threshold (see below) and at least one of the commands must not be involved in a lock contention that is handled by the ″hung″ command recovery.

The recovery action depends on the job definitions (see “ENQ section – resources to be monitored and jobs to be canceled” on page 53). If the job can be canceled, the recovery also removes its waiting commands and terminates its executing commands. The recovery action is completed either with message ING922E or with message ING924E. The latter message is repeatedly issued approximately every minute until the waiting queue becomes empty.

The threshold is calculated by subtracting the number of jobs that are issuing commands in the command class from the total number of TCBs (50) that are reserved for command processing. This prevents jobs that repeatedly issue few commands from being evaluated .

The recovery ends when the message IEE061I is issued.

Note: The dump definitions are not in effect if a dump should be taken when the job is canceled. This is because the recovery routine of the job that is being canceled can suppress the dump.

Customization

Automation of handling long-running enqueues is enabled by setting `ENQ` in the `AUTO` section of AOFCUST.

For details on the AOFCUST entries, see “Editing the customization member AOFCUST” on page 46.

Neither SYSIEFSD resource recovery nor hung command recovery need further customization.

System removal

The purpose of this function is to isolate failed systems from a Parallel Sysplex by removing them as quickly as possible. It also ensures fast mean time to recovery (MTTR) for those system images that you wish to restart immediately if an unavoidable outage occurs.

Note: This function is unavailable when running on a z/OS image which runs under z/VM®, even if the function is enabled.

In particular, the function automates the messages IXC102A and IXC402D.

The automation of the first message completes the Sysplex Failure Management (SFM). Under certain circumstances SFM cannot complete the isolation of a failed system. This is because SFM’s HW isolation, resetting the channel subsystem (CSS) of the failed system, is driven through the CF. When connectivity between the system image and the coupling facility is lost, SFM cannot perform the hardware isolation (ISOLATE command) and defers resetting the system image until manual operator intervention occurs. Message IXC102A tells the operator to manually reset the HW and then reply ″DOWN″ to the message, after which SFM safely partitions the system image out of the sysplex. The longer the delay lasts, the more the components and applications that rely on XCF messaging are impacted. The delay can eventually lead to a sysplex outage when the failed system has I/O operations pending. Automation of this message minimizes the delay.

The second message has the same impact as the first one. However, this message indicates a possible temporary inoperative status of the system due to a missing status update. For this reason the automation gives the system the chance to recover before the removal takes place by replying ″INTERVAL=sss″ to the first occurrence of message IXC402D. The interval time is calculated as twice the SPINTIME value (defined in parmlib member EXSPAT*xx*) plus 5 seconds.

The automation does the removal of a system in two stages. The first stage clears any pending I/O operations by sending a hardware command to the Support Element. This requires information about the software running on the hardware. Because the system issuing message IXC102A or IXC402D does not necessarily have access to the hardware of the failed system, the automation needs predefined mapping between software and hardware. Depending on this mapping, it then routes the hardware command to the system that has access to the hardware of the failed system. For further information about hardware requirements refer to “Preparing the hardware” on page 31 and “IXC102A section – hardware commands” on page 71.

The second stage replies to the outstanding WTOR with ″DOWN″ triggering the removal of the system from the sysplex.

Customization

Automation of message IXC102A is enabled by setting XCF in the AUTO section of AOFCUST.

For details on the AOFCUST entries, see “Editing the customization member AOFCUST” on page 46.

Recovering auxiliary storage shortage

With the automation of local page data sets, msys for Operations prevents auxiliary storage shortage outages by dynamically allocating spare local page data sets when needed. The function checks which jobs cause the shortage condition and whether additional page data sets can be added. If this is not possible, the job that is causing the shortage will be canceled if this has been defined.

To enable local page data set automation, you should customize the PAGTOTL parameter (defined in one of the IEASYS*xx* PARMLIB members used during IPL). Make sure that you set the PAGTOTL parameter to a value greater than the number of local page data sets currently used.

Local page data sets must be defined in the master catalog and should not be SMS-managed. It is recommended that you use pre-allocated local data sets instead of dynamically allocated ones. This makes the processfaster because formatting newly allocated page data sets is time-consuming (10sec./35MB). Each predefined local page data set should be allocated with 10% space of local page space that is currently used by the system. If predefined page data sets can no longer be allocated, new local page data sets will be created dynamically.

Customization

Automation of the recovery of auxiliary storage shortage is enabled by setting PAGE in the AUTO section of AOFCUST.

For details on the AOFCUST entries, see “Editing the customization member AOFCUST” on page 46.

Managing coupling facilities

A *coupling facility* (CF) is a logical partition that provides storage for data exchange between components of an application that is distributed across different systems in a Parallel Sysplex. A Parallel Sysplex can contain more than one CF. The storage of a coupling facility is divided into areas that are called *structures*. You can imagine a structure as a special kind of data set. It is these structures, which are identified by their name, that are accessed for reading and writing by the application components.

The association between CFs and structures is dynamic. A structure that is used by an application need not be allocated at all (for example, when the application is not running), and can be allocated on different CFs at different points in time. For every structure, there exists a *preference list* that defines the CFs on which it may be allocated. The order of the CFs in that list determines which CF is selected when more than one member of the list satisfies all allocation requirements (for example, provides enough space).

The preference list, the space requirements, and other properties of the structures are defined in the active CFRM policy. This policy is contained in the CFRM couple data set. Refer to “Managing couple data sets” on page 9 for further information.

XES allocates a structure that does not yet reside on any CF when an application component needs to be connected to it. Note that the application component only specifies the name of the structure that it wants to access. It is XES that decides on which CF the structure is allocated. This decision is influenced by the structure definition in the active CFRM policy. After the structure has been allocated, the requesting application component can access it, and further components of this application can require to connect to it. An application component that has access to an allocated structure is referred to as an *active connector* to this structure.

In the simplest case, XES deallocates a structure when all connected application components have disconnected from the structure. However, an application component can require that the structure or its own connection to the structure be *persistent*. When the *structure* is persistent it remains allocated even when the application component is no longer connected to it. When a *connection* is persistent the structure remains allocated after a failure of that connection. The application component in question remains a connector to the structure, although not an active one. It is now a *failed persistent* connector. In both cases, you can force the deallocation of the structure as soon as it no longer has active connectors.

Allocated structures can be *rebuilt*. Rebuilding is the process of reconstructing a structure on the same, or another, CF. A rebuild consists of three main steps. First, XES allocates the new structure instance. Then, the data of the old structure is reconstructed in the new structure. Finally, XES deallocates the old structure instance. Note that you cannot specify the target CF in your rebuild request. As with structure allocation, XES selects it from the preference list.

There are two methods for rebuild: user-managed, and (from OS/390 2.8 onward) system-managed. With user-managed rebuild, the active connectors are responsible for reconstructing the data. With system-managed rebuild, XES transfers the data to the new structure instance. System-managed rebuild is thus also available for structures without active connectors. These structures can either themselves be persistent or have failed persistent connections.

When an application component connects to a structure, it specifies whether it allows the structure to be rebuilt through user-managed or system-managed rebuild. For structures with active connectors, both rebuild methods require that all active connectors allow the respective rebuild method.

You can also *duplex* structures. Duplexing means maintaining two instances of the same structure on different CFs at the same time. Duplexing serves to increase availability and usability of a structure.

Typical management tasks for CFs are removing a CF from the sysplex and reintegrating it again. These tasks have several steps that must be performed in a certain order and can be quite complex. To simplify these operations, msys for Operations offers the INGCF command. INGCF has several functions, which serve to manipulate structures and the CFs themselves. These functions are briefly described in the following. For more information, see “INGCF” on page 134.

Some functions deal with the sender paths of a coupling facility. They have the following limitations. First, at least one system in the sysplex that is running the automation must know the control unit id (CUID) of the coupling facility. If this is not the case, no missing sender paths can be resolved.

A missing sender path occurs when a coupling facility is deactivated prior to a system IPL (or reIPL) and then activated afterwards. The system that has been IPLed (or reIPLed) does not recognize the coupling facility. To determine the missing sender paths, the automation calls the HOM interface of HCD. Resolving the missing path information is only possible when either the complete network address is defined in HCD along with the processor id, or you provide the CPC synonym used by the automation as the processor id. However, it is recommended that you define both. If neither is defined, the system that misses the sender paths must run the automation.

INGCF DRAIN

INGCF DRAIN displays information about the allocated structures of a CF and supports removal of this CF from the sysplex. Usually, draining a CF requires that at least one alternate CF be enabled for the sysplex.

With INGCF DRAIN, you can perform the following sequence of tasks:

1. Rebuild all structures that can be rebuilt with user-managed or system-managed rebuild on an alternative coupling facility, and deallocate structure instances on the target CF that are being duplexed on another CF. For duplex structures, the duplexing process is stopped.

   The scope of the rebuild action depends partly on the release level of the systems from which the structures were allocated:

   - Structures that were allocated from a system with OS/390 2.7 or below can only be rebuilt if they have at least one active connector and all its active connectors support user-managed rebuild.
   - Structures that were allocated from a system with OS/390 2.8 or above can be rebuilt if they have an active connector and support either user-managed or system-managed rebuild, or if they have no active connector.

   Note: INGCF DRAIN rebuilds structures one at a time (SETXCF START,REBUILD,STRNAME=), not globally (SETXCF START,REBUILD,CFNAME=), and always on a CF that is different from the target CF (LOCATION=OTHER).
2. Force the deallocation of structures that have no active connectors and could not be rebuilt.
3. Disconnect the coupling facility from the systems to which it is connected.
4. Deactivate the coupling facility.

   Note: This task is unavailable when running on a z/OS image which runs under z/VM.

INGCF DRAIN ensures that the supported actions are carried out in the right order. Thus, for example, INGCF DRAIN lets you disconnect the coupling facility from the systems only after all structures of the coupling facility have been moved to another CF or have been deallocated. After each step, INGCF DRAIN presents the results of that step. You can then choose whether you want to initiate the next step.

For further information about the INGCF DRAIN command refer to “INGCF” on page 134.

INGCF ENABLE

INGCF ENABLE is the counterpart of INGCF DRAIN. It supports integration of a new CF into a sysplex and reintegration of an existing CF into the sysplex, for example, after maintenance of the CF.

Note: INGCF ENABLE assumes that the receiver paths from the CF to the systems in the sysplex have been defined and activated. This requires a POR of the CPC on which the CF resides.

With INGCF ENABLE, you can perform the following sequence of tasks:

1. Activate the coupling facility.

   Note: This task is unavailable when running on a z/OS image which runs under z/VM.
2. Connect the systems of the sysplex with the coupling facility (sender paths).
3. Switch to another CFRM policy if
   - the target CF is not defined in the active policy, and
   - a policy is available that contains the target CF and definitions for all active CFs and all allocated structures.
4. Populate the target CF, that is, rebuild all those structures on the target CF, if this CF is the first usable one in the preference list, provided that this is not excluded by other requirements.

When the structures have been allocated on the target CF, INGCF ENABLE displays the result.

As with INGCF DRAIN, INGCF ENABLE ensures that the supported actions are carried out in the right order. Thus, you can only start populating the target CF after it has been connected to the systems in the sysplex.

For further information about the INGCF ENABLE command refer to “INGCF” on page 134.

INGCF PATH

INGCF PATH lets you set the sender paths ONLINE or OFFLINE. The last sender path can only be set offline when no more structures are allocated. For further information about the INGCF PATH command refer to “INGCF” on page 134.

INGCF STRUCTURE

INGCF STRUCTURE displays all the allocated structures of a CF and information about their current conditions. For a selected structure, you can:

- Display detailed information
- Initiate a rebuild on another CF, depending on the rebuild pending status (PENDING calls LOCATION=NORMAL, otherwise LOCATION=OTHER)
- Force the deletion of the structure
- Start and stop duplexing

Rebuild and deletion can only be performed for structures with certain conditions.

For further information about the INGCF STRUCTURE command refer to “INGCF” on page 134.

Customization

None. For information on how to control access to INGCF, refer to “Command authorization” on page 77.

Note that the ENABLE function requires that the active IODF is catalogued. Otherwise, sender path information cannot be retrieved in certain situations.

The IBM Health Checker for z/OS and Sysplex

The IBM Health Checker for z/OS and Sysplex is a tool that checks the current, active operating system (z/OS or OS/390) and sysplex settings and definitions for an image, and compares their values to those either suggested by IBM or defined by you, as your criteria. The objective of the HealthChecker is to identify potential problems before they impact your availability, or in worst cases, cause outages. The function produces reports (snapshots of your system) to help you analyze the values defined for this system. msys for Operations can automate the running of the checks **sysplex-wide** and provides an easy-to-use interface for viewing the report data.

General prerequisites

The following operating systems are supported by the HealthChecker:

- OS/390 R10 or later
- all z/OS releases

The following system configurations are supported:

- XCFLOCAL is NOT supported
- MONOPLEX
- MULTISYSTEM

HealthChecker best practice values

The values used by the HealthChecker are also referred to as best practices and originate from a variety of sources, including books and Web sites. However, the fact that the information comes from various sources can make it more difficult for you to ensure that your configurations reflect all of the suggestions. Using the HealthChecker means that this work is done for you. Another problem is keeping up with the changes that may have been made on your systems and ensuring that they still reflect either IBM’s suggestions or your own criteria. To address this, you can ensure that the HealthChecker be run on demand or automatically, and hence easily determine if new values have introduced potential exceptions. We also realize that there are customer-unique and system-unique cases where the IBM suggestions are not appropriate. Therefore, you can either specify overrides to IBM values or suppress the running of a check. See “HEALTHCHK section – specifying user overrides” on page 57 for details.

The HealthChecker checks the current values that are being used by your system; it does not check PARMLIB values. The scope of the checks are the local system where the function is run. It does not check values on other systems within the sysplex, although some values checked are sysplex-wide in scope. We recommend that you run the HealthChecker on all systems in your sysplex. In this case, all the systems in your sysplex will run the LOCAL checks (system-wide scope) but only one system in your sysplex will run the GLOBAL checks (sysplex-wide scope) in addition to the LOCAL checks. The way this latter system is determined is such that the HealthChecker function does an exclusive ENQ on a global GRS resource. The system that gets that LOCK will also do the Global checks.

When the HealthChecker function is enabled, it performs regular checks at predefined time intervals. The time intervals are defined individually for each check as part of IBM's best practices, although you can also override them. The checks are done based on IBM's best practices or your overrides. The HealthChecker implements the best practices in these ways:

1. Consolidates best practice values from multiple IBM sources
2. Reports on your configuration's active settings compared to IBM's suggestions, simplifying administration and operations
3. Reports on your configuration's active settings specific to any customer-specified preferences that can be used to override IBM values
4. Provides a mechanism for IBM to distribute updates to best practice values or to provide additional checks in a manner that is easily integrated into your environment

The basis for the values used by the IBM Health Checker for z/OS and Sysplex include:

- Parallel Sysplex and z/OS publications:
  - *z/OS MVS Setting Up a Sysplex*
  - *z/OS MVS Planning: Operations*, SA22-7601
  - *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- Parallel Sysplex Availability Checklist

  The Parallel Sysplex Availability Checklist can be found at:

  `http://www.ibm.com/servers/eserver/zseries/pso/`
- ITSO Redbooks
  - *OS/390® Parallel Sysplex Configuration, Volume 1: Overview*, SG24-5637
  - *OS/390 Parallel Sysplex Configuration, Volume 2: Cookbook*, SG24-5638
  - *OS/390 Parallel Sysplex Configuration, Volume 3: Connectivity*, SG24-5639

  The Redbooks™ can be found at:

  `http://www.redbooks.ibm.com/`
- *z/OS Parallel Sysplex Test Report*, SA22-7663

  The Parallel Sysplex Test Report can be found at:

  `http://www.ibm.com/servers/eserver/zseries/zos/integtst/`
- Washington System Center Flashes

  Washington System Center Flashes can be found at:

  `http://www.ibm.com/support/techdocs/`

  Of particular interest for migration to a 64-bit environment is whitepaper WP100269, *z/OS Performance: Managing Processor Storage in a 64-bit environment*, and Washington System Center Flash 10086, *Software Capacity Planning: Migration to 64 bit Mode*.

INGPLEX BESTpractices

This command allows you to view the currently active best practices (for a description of the command, see "INGPLEX" on page 163).

INGHC

This command has two purposes:

1. Display the results of the checks
2. Trigger actions in the HealthChecker

Types of reports

The HealthChecker reports reflect values at a point in time (snapshot). The report is comprised of a series of records in the System Logger. These records are comprised of the following components:

- Message text and explanation
- Actions that can be taken in case of an exception to address the exception
- IBM suggestions
- Reasons for IBM's suggestions

Types of actions

The following types of actions are available:

- You can request selected or all checks to be done immediately instead of waiting for the time interval of the respective checks to elapse. This is useful if you have made some change to your system and you want to immediately have these changes checked against the best practices. You can select individual systems in your sysplex or all at once.
- When you define or change your overrides to IBM's best practices, you can request the HealthChecker to take notice of these updates. You can select individual systems in your sysplex or all at once. The latter is useful in particular if the AOFCUST member that contains your overrides is shared among all of your systems.

Customization

Automation of the HealthChecker is enabled by setting `HEALTHCHK` in the `AUTO` section of AOFCUST.

You can override IBM's suggestions in the `HEALTHCHK` section in the AOFCUST member in order to:

- Specify your own values for a check
- Disable the running of a check

Hardware validation

This function performs cross-validation of the hardware configuration mapped out in AOFCUST against the actual hardware configuration that is running. This information is critical to accurately control logical partitions (LPARs) on any supported CPC within the HMC/SE LAN over the BCP Internal Interface.

Hardware validation uses the CPC name, Partition name and Partition number to ensure that the LPARs defined in AOFCUST are on the correct CPC and located on the correct partition number. However, this helps only for coupling facilities because their partition identifiers must be defined in the active CFRM policy.

For MVS images, information from the HMC/SE (such as system name and sysplex name that are stored during initialization) is used to verify the corresponding AOFCUST definitions. During initialization of the automation's Hardware Command Interface and just before a disruptive request is sent to a partition, new checks are made to ensure that everything matches correctly.

Note: Only active images can be verified. For inactive images we must still rely on definitions made in AOFCUST.

An active system in this context is a system belonging to the same sysplex as the system that runs the hardware validation, that is msys for Operations checks only systems and coupling facilities within its own sysplex.

Hardware validation runs on an msys for Operations system primarily during startup, and subsequently when changes to the definition in AOFCUST are applied through the ACF command (ACF COLD). The validation checks the definitions of all registered systems, that is whenever an msys for Operations system performs the hardware validation, it validates all systems and coupling facilities that are active in the sysplex at this point in time. Registered systems are systems running msys for Operations or SA OS/390 that have joined the same XCF group.

The validation of active systems and coupling facilities requires that the CPCs that host the active systems must all be defined in the HW section of AOFCUST.

The data for inactive systems cannot be verified. However, these definitions are checked for consistency across all registered systems. As soon as one of these inactive systems or coupling facilities joins the sysplex or is made available for use, the validation is run for the particular image only.

Retrieving actual hardware information can take up to 5 minutes per CPC depending on the model and its LPARs. During the time that the hardware validation takes place all other hardware-related automation is either delayed or cannot be performed, depending on the type of recovery. For this reason the validation carries out ″delta″ processing. That is validating only the data that has changed. This also includes the absence of data resulting in terminating CPC connections when CPC definitions are missing that have been applied by a prior validation. The actions resulting from the validation are performed on ALL registered systems. This has two advantages:

- you don’t need to recycle NetView for changes in hardware definitions.
- you only need to make the changes available to one system.

The first part of the hardware validation triggered by the ACF command or the automation startup determines what CPC connections must be terminated and initiated, namely in this sequence. The resulting actions are performed on all registered systems. When this step has been completed successfully the image validation is performed.

The image validation collects actual hardware information, and verifies the current hardware definitions against the actual data and the definitions found on all other registered systems. It informs you if:

- a real system or coupling facility could not be validated because either actual hardware information or user definitions are not available
- the image definitions could not be evaluated because the actual hardware information is not available
- the real system or coupling facility is not active and the image definitions of some of the registered systems are different
- any definition value has been corrected that was improperly defined or not defined at all

Changes in hardware definitions can be made available to all registered systems by simply invoking the command ACF COLD at only one of the these systems. There is one exception: the change of the authorization token value used for the communication with a particular CPC. A change of this value requires 3 steps:

1. In the first step you must remove the particular CPC definition and then invoke the ACF command as above.
2. When the command completes successfully the next step is to change the authorization token value of the CPC at the Support Element.
3. The final step is to define the CPC again with the new token value and invoke the ACF command again.

Note: This behavior of the ACF command applies to the hardware definitions ONLY.

The second part of the validation is triggered by either the message IXC517I that is issued when a coupling facility is made available for use, or by the automation itself when notified that a system joined the sysplex. Both trigger the automation to perform only the validation of the new system or coupling facility. Multiple occurrences of messages for the same system or coupling facility are ignored while this system or coupling facility is validated. In case of a new system, the advantage here is that the real hardware is validated before the system starts NetView and the automation. If this automation then detects no difference between its current definitions and the definitions of the other registered systems—which is the normal case—only a consistency check takes place. This check does not require any real hardware information.

Prerequisites

Hardware validation has the following prerequisites:

- msys for Operations must have been initialized. For this reason the validation is delayed until the initialization has completed.
- All coupling facilities that are used in the sysplex must reside on a CMOS-S/390 G5 processor or higher. Only these processors return the partition identifier that is required for validating coupling facilities.
- The BCP Internal Interface must have been initialized to accept requests. Or, when unavailable, at least one other registered system must have access to the hardware. Registered systems are systems running msys for Operations or SA OS/390 that have joined the same XCF group.

Note: Hardware validation is not supported on MVS systems running under z/VM.

Miscellaneous sysplex functions

Recording IPL Information

With the INGPLEX IPL command you can record, view and compare the operating system's IPL information. If, after the IPL, a system does not behave as expected, the IPL recording function enables you to identify parameters that were changed, for example, since the last IPL. The recording function enables you to compare different IPL scenarios. INGPLEX IPL is a tool that helps to identify and resolve the cause of startup problems. For further information about the INGPLEX IPL command refer to Chapter 14, "Sysplex-related commands," on page 133.

System dump options

The enhanced INGPLEX functions allow you to control dump options sysplex-wide, for *registered* systems, that is, those on which the automation runs. For further information refer to Chapter 14, "Sysplex-related commands," on page 133.

Multisystem dumps

One of the enhanced INGPLEX functions provides an easy-to-use interface for multisystem dumps. For further information refer to Chapter 14, “Sysplex-related commands,” on page 133.

SLIP traps

The enhanced INGPLEX functions also let you view, enable, disable, and delete SLIP traps defined in the sysplex. For further information refer to Chapter 14, “Sysplex-related commands,” on page 133.

Part 2. Setting up msys for Operations

This part describes how to install and customize msys for Operations. It has the following chapters:

Chapter 3. Installing msys for Operations

This chapter describes the prerequisites and the steps to follow in planning and preparing for the installation of msys for Operations.

Functional prerequisites

To obtain current service recommendations and to identify current product service requirements, contact the IBM Customer Support Center or use S/390® SoftwareXcel to obtain the current ″PSP Bucket″.

The following lists the functional hardware and software prerequisites that are required for using the enhanced Parallel Sysplex automation functionality shipped with APARs OW50146 and OW56107.

Software prerequisites

The following APARs need to be installed:

APAR	PTF	FMID	Area	Function
OW56013	UW94365	HBB7703	msys	Retrofit for OS/390 R10 and z/OS V1R1

This APAR enables msys for Operations to run on OS/390 R10 and z/OS V1R1. This APAR also addresses the following problems:

- problems with migration to SA OS/390 V2 due to syntax errors in the generated ACF fragment Z999APRO
- the message `AOF784I AUTOMATION CONTROL IS EMPTY OR DOES NOT EXIST` appears during initialization
- during WTO(R) buffer shortage recovery, `NOVALUE CONDITION TRAPPED IN INGRX735 LINE 97` occurs
- APPL START UP PROC sample INGNVAP0 contains outdated information

APAR	PTF	FMID	Area	Function
OW51923	UW84343 UW84344 UW84345	HBB7703 HBB7705 HBB7706	SPI	BCP (Basic Control Program) internal interface used by system recovery and coupling facility functions

This APAR allows multiple applications (especially HCD and msys for Operations) to use the BCP Internal Interface simultaneously. msys for Operations uses this interface to activate, inactivate, and query a coupling facility as well as send hardware commands to the LPAR of the system which is being partitioned out of the sysplex.

APAR	PTF	FMID	Area	Function
OW52369	UW86446 UW86444 UW86445	HBB7703 HBB7705 HBB7706	XCF	Coupling facility functions

With z/OS 1.2, XCF introduced a different behavior from the previous functionality of how users are informed when a rebuild duplex process has completed. This also applies to OS/390 R10. 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. This APAR resolves the situation by informing the user who started the duplex process as well as the user who stops it.

APAR	PTF	FMID	Area	Function
OW56587	UW94992	HBB7703	GRS	Elimination of long running ENQs
	UW94993	HBB7705		
	UW94994	HBB7706		
	UW94995	HBB7707		

The automation of eliminating long-running ENQs is only available when this APAR is installed. After installing the APAR 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 the APAR has been applied. Recycling the NVSS address space has the same result.

APAR	PTF	FMID	Area	Function
OW53637		HBB7703	XCF	Structure rebuild function
		HBB7705		
		HBB7706		

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. This also applies to OS/390 R10. This also affects messages issued on behalf of the SETXCF ALTER command. When the automation rebuilds a structure, it also checks its initial and current size. If the initial size is less than the current size, the automation tries to change the initial size to the current 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. This APAR resolves the problem.

Hardware prerequisites

Note: msys for Operations does not support any hardware commands when running on a z/OS image which runs under z/VM. Refer to Chapter 2, “Sysplex functions,” on page 9 for information about which particular functions are affected.

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.

The following sections describe how to install and customize msys for Operations. It is assumed that you have completed all the steps that are described in the program directory, and that the target libraries for both the Sysplex Functions and the NVSS part of msys for Operations are available.

If you are using SA OS/390 V1R3 and migrate from OS/390 R9 and prior releases to OS/390 R10 and later releases, you can enable msys for Operations functions and run SA OS/390 V1R3 on the same systems. For details, see Appendix C, "Coexistence of msys for Operations and SA OS/390 releases," on page 265. If you are using SA OS/390 V2R1, you must install PTFs UW90711 and UW90712 (APARs OW37539 and OW48837) and APAR OW56013, which provide all the functions of msys for Operations within SA OS/390 V2R1, so that there is no need to install msys for Operations.

The installation of msys for Operations has the following steps:

- Preparing the hardware
- Planning for domain and console names
- Preparing the MVS System
- Preparing the msys for Operations data sets
- Defining msys for Operations to VTAM
- Updating the start procedure and the initialization style sheet
- Editing the AOFCUST member with which the Sysplex Functions are customized

msys for Operations must be installed and run on every MVS image of the Parallel Sysplex you want to control or retrieve information from. The customization effort can be minimized by sharing the data sets with which msys for Operations is installed and through the use of symbols where systems require certain unique values. This can be done by using system symbols that are resolved differently on every system.

Note: When MVS is running under z/VM, msys for Operations does not support any hardware commands. See Chapter 2, "Sysplex functions," on page 9 for more details.

Preparing the hardware

This section describes the steps necessary to prepare your processor hardware in the sysplex to use the Parallel Sysplex enhancements shipped with APAR OW50146.

Understanding the hardware interface

In order to allow the sysplex-wide activation or deactivation of the coupling facilities and to control sysplex members leaving the sysplex, msys for Operations 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 MVS BCP Internal Interface allows you to send hardware operations commands, such as SYSTEM RESET or ACTIVATE, to the Support Element attached to the individual processor hardware (CPC). If the CPC is configured in LPAR mode, the operations command can be sent to all logical partitions defined to the CPC.

With the enhanced sysplex functions of msys for Operations, sysplex members running on CPCs other than their own image can also be controlled through the BCP Internal Interface. This is possible by defining all CPCs in your sysplex to 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 msys for Operations BCP Internal Interface to control itself or other processors:

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

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 msys for Operations to perform these tasks, nor can HCD be used to perform the hardware operations functions of msys for Operations.

Preparing the master Hardware Management Console

To prepare the master HMC, 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.

Select **Console Actions** and click on the *Enable Hardware Management Console Services* icon. Set the LIC change Enabled radio button. Press the OK button to save the change or press cancel if LIC change 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.

Preparing the Support Element

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 API and set the community name
- Set the cross partition flag (LPAR mode)

For additional SNMP and API configuration information please refer to *zSeries 900 Application Programming Interfaces*, Chapter 6, ″Configuring for the Data Exchange APIs″.

Configure SNMP

You have to specify two community names to use the BCP Internal Interface. For this task, you need to be logged on in *Access Administrator* mode on your CPC’s Support Element. To complete this task:

1. Start the SNMP Configuration task by double clicking the **Console Actions** icon in the *Views* area of the Console.

2. Select the **Communities** tab of the SNMP Configuration notebook window.
3. For the **API** community name, enter the following information and select the **Add** push button to add the new community name:

   Protocol
   Select UDP from the drop-down list.

   Name
   The API Community name you have chosen.

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

   Network Mask
   255.255.255.255

   Access Type
   Select the **Read only** radio button.
4. For the **BCP Internal Interface** community name, enter the following information and select the **Add** push button to add the new community name:

   Protocol
   Select UDP from the drop-down list.

   Name
   MSYSOPS (Use the CPC authtkn name that you defined for the CPC in the HW section of your hardware configuration entry of the AOFCUST)

   Address
   127.0.0.1

   Network Mask
   255.255.255.255

   Access Type
   Select the **Read/write** radio button.
5. Select the **OK** push button to save the changed settings and close the SNMP notebook window.
6. 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.

Enable the API and set the community name

In order to use the BCP Internal 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** check box.
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

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** check box.

Planning for the domain and EMCS console names

msys for Operations must be installed on every system of the sysplex. For each of these instances, you must specify a domain name, and that name must be unique across the network. Also, when you wish to run msys for Operations and a licensed NetView on the same system, this instance of NetView must have a domain name that is different from that of all the msys for Operations instances within the network.

Furthermore, the automation router task that receives MVS messages and passes them over to automation will be running within every instance of msys for Operations. For these tasks, you must define a name that is unique among all msys for Operations tasks across the sysplex. Moreover, when you run msys for Operations and a licensed NetView on the same system, the automation router task of the licensed product must also have a name that is different from those of all msys for Operations instances.

When planning a scheme for the domain and automation router names you might consider using system symbols for the following reasons:

- You can share certain data set members, for example, the msys for Operations start procedure (MSOAPROC), among all systems of the sysplex.
- You can add further systems to your sysplex, or install a licensed NetView on a system on which msys for Operations is running, without any changes to your naming conventions.

Domain names

Every instance of msys for Operations must be assigned a domain name that is unique across the network. The samples supplied with msys for Operations use `MSO&SYSCLONE` as the domain name. `&SYSCLONE` is a system-defined symbol whose default value is the last two characters of the system name, as specified for the `&SYSNAME.` symbol, which is also system-defined. You can override the default by specifying a value for `&SYSCLONE` in IEASYM*xx*.

You can retain this name for all instances of msys for Operations if the following conditions are satisfied:

- The value of `&SYSCLONE` is different for every system of the sysplex.
- You do not run an msys for Operations instance and a licensed NetView on the same system.

The domain name must be specified on one hand in the start procedure MSOAPROC of msys for Operations or the initialization style sheet (see “Preparing the startup procedure and the initialization style sheet” on page 44), and on the other hand in the VTAM definitions (see “Preparing VTAM” on page 43).

EMCS console names and the automation router task

Operator tasks use EMCS consoles to send commands from msys for Operations to the MVS operating system and to receive messages from MVS. The automation router task (see “Defining the name of the automation router task” on page 46) also uses an EMCS console to receive messages from MVS. You can explicitly assign an EMCS console to an operator task or let msys for Operations assign it automatically. The EMCS console for the automation router task is always assigned by msys for Operations.

When an EMCS console is assigned to a task it is given a name. If you assign the console explicitly, you must specify that name. If msys for Operations assigns the console automatically, it uses the name of the task as the console name.

The important point is the EMCS console names must be unique across the sysplex, and that it is difficult to meet this requirement when the console names are identical with the task names. When, for example, operator A has logged on with the ID OPER1 on system KEY1, and has been assigned an EMCS console by msys for Operations, this console has the name OPER1. If operator B logs on subsequently on system KEY2 with the same operator ID, the automatic assignment of an EMCS console for the OST on KEY2 will fail, because a console with the name OPER1 already exists within the sysplex. B will not be able to issue MVS commands.

For operator tasks, you can avoid such a name clash by specifying the command list LOGPROF1 as the initial command in the operator profile (IC parameter in the NetView segment of RACF). This command list, which is supplied by msys for Operations, will then assign an EMCS console with a unique name to the operator task. The characters that are used in determining unique console names can be tailored by updating the common global variable AOFCNMASK. This global is used as a hex mask to extract characters from the following string when generating unique console names with command AOCGETCN.

```
left(opid(),8)||right(opid(),8),
||left(aofsysname,4)||right(aofsysname,4),
||left(applid(),8)||right(applid(),8),
||'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$&#155;#@_|?'
```

where:

opid()
is a function which returns the OST task name

aofsysname
is a common global which stores the system name

applid()
is a function which returns the VTAM LU name

The default for AOFCNMASK is 290C0D0E0F101718. X'29' selects character A in position 41, X'0C' to X'10' selects the last five characters of the opid in positions 12 to 16, and X'17' and X'18' select the last two characters of the sysname in positions 23 and 24. If AOFCNMASK is null, AOCGETCN will attempt to obtain a unique extended MCS console after a one-minute interval, followed by a two-minute interval and so forth for a maximum of 5 passes (15 minutes elapsed from the initial invocation of the command).

The AOFCNMASK common global can be set using the CNMSTYLE set COMMON global function. For example:

```
COMMON.AOFCNMASK = 2A01020304051718
```

X'2A' selects character B in position 42, X'01' to X'05' selects the first five characters of the opid in positions 1 to 5, and X'17' and X'18' select the last two characters of the sysname in positions 23 and 24. This entry should be placed adjacent to the COMMON.WAITTIME entry in your C*xx*STYLE version of CNMSTYLE. For this entry to be activated msys for Operations will need to be restarted. For more details, see “Defining operator logon attributes in the NVSS segment of an SAF product” on page 76.

Planning for the domain and EMCS console names

As for the automation router tasks of the individual msys for Operations instances, the only way to avoid naming conflicts of the associated EMCS consoles is to ensure that the task names are unique sysplex wide. The name of the automation router task for the individual msys for Operations instances is specified in the initialization style sheet. For details, see "Defining the name of the automation router task" on page 46.

Target libraries

Table 1 shows a list of target data sets as provided by the SMP/E installation process to be used for production on your system.

Table 1. Target data sets

Data Set Name	Description
NETVIEW.VnRnMn.BNJPNL1	NVSS panels
NETVIEW.VnRnMn.BNJPNL2	NVSS panels
NETVIEW.VnRnMn.BNJSRC1	not used
NETVIEW.VnRnMn.CNMCLST	NVSS clists
NETVIEW.VnRnMn.CNMINST	NVSS installation
NETVIEW.VnRnMn.CNMLINK	NVSS program library
NETVIEW.VnRnMn.CNMPNL1	NVSS online help
NETVIEW.VnRnMn.CNMSAMP	NVSS samples
NETVIEW.VnRnMn.DSIPARM	NVSS definition members
NETVIEW.VnRnMn.DSIPRF	NVSS operator profiles
NETVIEW.VnRnMn.NVULIB	NVSS program library
NETVIEW.VnRnMn.SCNMLNK1	NVSS link list modules
NETVIEW.VnRnMn.SCNMLPA1	NVSS LPA modules
NETVIEW.VnRnMn.SCNMMAC1	NVSS macros
NETVIEW.VnRnMn.SDSIMSG1	NVSS messages
NETVIEW.VnRnMn.SDSIOPEN	NVSS key settings
NETVIEW.VnRnMn.SDUIMSG1	not used
NETVIEW.VnRnMn.SEKGCAS1	not used
NETVIEW.VnRnMn.SEKGLANG	not used
NETVIEW.VnRnMn.SEKGLNK1	not used
NETVIEW.VnRnMn.SEKGLUTB	not used
NETVIEW.VnRnMn.SEKGMOD1	not used
NETVIEW.VnRnMn.SEKGMOD2	not used
NETVIEW.VnRnMn.SEKGPNL1	not used
NETVIEW.VnRnMn.SEKGSMP1	not used
NETVIEW.VnRnMn.SEZLCLST	not used
NETVIEW.VnRnMn.SEZLINST	not used
NETVIEW.VnRnMn.SEZLLINK	not used
NETVIEW.VnRnMn.SEZLPNLU	not used
NETVIEW.VnRnMn.SEZLSAMP	not used
NETVIEW.VnRnMn.SFLBDAT1	not used

Table 1. Target data sets (continued)

Data Set Name	Description
ING.SINGMOD1	Sysplex Functions modules
ING.SINGMOD2	Sysplex Functions modules
A ING.SINGMOD3	LPA modules
ING.SINGNMSG	Sysplex Functions messages
ING.SINGNPNL	Sysplex Functions panels
ING.SINGNPRF	Sysplex Functions profiles
ING.SINGNPRM	Sysplex Functions DSIPARM and other samples
ING.SINGNREX	Sysplex Functions REXX execs
ING.SINGFautoSAMP	Sysplex Functions general samples

Preparing the MVS system

This chapter describes the steps necessary to prepare your MVS system for the installation of msys for Operations.

Modifying the maximum number of language processor (REXX) environments for msys for Operations

Before the TSO/E language processor can process an exec, a language processor environment must exist. A language processor environment is the environment in which a REXX exec runs. The following discusses how msys for Operations uses these REXX environments and highlights issues to consider when estimating the number of language processor environments needed for your configuration.

msys for Operations provides several parts that contain REXX source code.

msys for Operations also contains several parts that make use of the Data REXX function. The Data REXX function allows the inclusion of REXX instructions and functions in data files.

When a REXX command list is run in msys for Operations, the REXX interpreter sets up a language processor environment for msys for Operations. When the command list ends, this unique environment can be held for reuse by the same task. msys for Operations retains these REXX environments to improve REXX environment initialization performance. As a result, it is very important to have a sufficient number of REXX environments available to msys for Operations. If more blocks are required than are available, msys for Operations issues the CNM416I REXX environment initialization error message.

msys for Operations retains up to three REXX environments and their associated storage until the operator logs off. Additionally, msys for Operations will *always* retain one REXX environment per task for Data REXX use.

The IRXANCHR table is a Time Sharing Option Extensions (TSO/E) table used to reserve storage for REXX environments. Both msys for Operations and TSO/E refer to this table when allocating storage for each REXX environment that is activated.

To change the number of environment table entries, you can use the IRXTSMPE sample that TSO/E provides in SYS1.SAMPLIB or you can create your own IRXANCHR load module. The IRXTSMPE sample is a System Modification Program/Extended (SMP/E) user modification (USERMOD) to change the number of language processor environments in an address space. The prologue of IRXTSMPE has instructions for using the sample job. The SMP/E code that is included in the IRXTSMPE sample handles the installation of the load module.

Note: IBM recommends that you set the number of environments to 200. This would require to set the `ENTRYNUM` parameter in the call of IRXANCHR call to 401.

Updating member SCHED*xx*

Define the msys for Operations program as nonswappable in MVS. The msys for Operations program must run in MVS storage key 8.

To make the msys for Operations program nonswappable, use the SCHED*xx* member of SYS1.PARMLIB. Ensure the SCHED*xx* statement for the msys for Operations program is PGM=DSIMNT. You should use the sample INGSCHE0 supplied with ING.SINGSAMP.

Updating member MPFLST*xx*

In order to enable message automation with msys for Operations, you must code statements for the relevant messages in the PARMLIB member MPFLST*xx* (see “Automatic recovery and message automation” on page 6). The INGEMPF member which resides in the Sysplex Functions sample library ING.SINGSAMP contains all the necessary statements. Copy these statements into your MPFLST*xx* member. Review the MPFLST*xx* member to ensure that it is appropriate for your system and resolve any conflicts. Review the MPFLST*xx* member to ensure that it is appropriate for your system and resolve any conflicts.

`AUTO(YES)` is required in the `.NO_ENTRY` statement 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 message automation table also specified in MPFLST*xx* with `AUTO(YES)`, you can specify `AUTO(NO)` for the `.NO_ENTRY` statement.

Message Automation

Messages processed by the automation either via the NVSS MAT (message automation table) or by the NVSS 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 IEA232I 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 IXC263I IXC309I
```

```
IXC402D IXC500I IXC501A IXC517I IXC559I
IXC560A
IXG257I IXG261E
IXL126I IXL127A
```

Updating the link list, LPA, and APF authorizations with PROG*xx*

The member INGPROG0 of ING.SINGSAMP contains a sample PROG*xx* member. Use this member to update the link list, LPA, and the APF authorizations dynamically. In order to activate the updates, follow the instructions in the sample.

Note: The dynamic updates are only in effect until the next IPL. In order to make them persistent, add a line to your COMMND*xx* member as follows:

```
COM='SET PROGxx'
```

Updating member COMMND*xx*

Define the msys for Operations procedure collecting the IPL information in MVS. Add the following statement to a COMMND*xx* Parmlib member that is shared by all systems in the sysplex:

```
COM='S HSAPIPLC,SUB=MSTR'
```

Create member CTIHSAZZ

Copy the CTIHSAZZ member residing in the SINGSAMP sample library into SYS1.PARMLIB.

Preparing msys for Operations

This chapter describes how to:

- Allocate the user data sets and VSAM clusters required by msys for Operations
- Load required members into user and system data sets
- Defining the logger CDS samples in ING.SINGSAMP

Allocating data sets and VSAM clusters using job INGALLC0 and INGALLC4

The job INGALLC0 is contained in member INGALLC0 of ING.SINGSAMP. It allocates partitioned data sets and VSAM clusters. They are listed, and their function is briefly explained, in Table 2 on page 40 and Table 3 on page 40. The data set and cluster names used in the tables are those that are supplied with the sample job; '*xxxxx*' is taken as a place holder for the domain name. The third column specifies the DD statement of the msys for Operations startup procedure MSOAPROC where the data set or cluster is be specified (see "Modifying the msys for Operations startup procedure" on page 44).

The job INGALLC4 is contained in member INGALLC4 of ING.SINGSAMP. It allocates the IPLDATA VSAM cluster also listed in Table 3 on page 40. The initial record in IPLDATA can be customized to control the maximum number of IPL records per system that will be stored and whether comments in PARMLIB members are to be stored. The data set must be allocated on a volume shared by all systems in the sysplex. Customize and run the JCL to allocate the data set. Recatalog the data set on all other systems in the sysplex if the catalog is not shared.

The PDSs are listed in the following table.

Table 2. Partitioned data sets for msys for Operations

Data Set Name	Function	DD Statement
MSOPS.USER.DSIPARM	Will hold shared definition members, for example the AOFCUST (see "Editing the customization member AOFCUST" on page 46).	DSIPARM
MSOPS.USER.*xxxxx*.DSIPARM	Will hold system–specific definition members.	DSIPARM
MSOPS.USER.*xxxxx*.DSILIST	See note 1.	DSILIST
MSOPS.USER.*xxxxx*.DSIASRC	See note 1.	DSIASRC
MSOPS.USER.*xxxxx*.DSIARPT	See note 1.	DSIARPT
MSOPS.USER.*xxxxx*.VTAMLST	Will contain the VTAM source definitions. See "Preparing VTAM" on page 43.	DSIVTAM
MSOPS.USER.VTAMLIB	Will contain VTAM load modules. See note 2.	

Notes:

1. This data set is not required for msys for Operations. However, this data set is required for NetView operation. It is strongly recommended that you allocate this data set to enable installation of the NetView product in the future.
2. Add the VTAMLIB data set to the list of authorized libraries in the IEAAPF*xx* of SYS1.PARMLIB.

The following table contains the VSAM clusters:

Table 3. VSAM clusters for msys for Operations

Data Set Name	Function	DD Statement
NETVIEW.VnRnMn.*xxxxx*.DSILOGP	Contains the primary network log. See Chapter 11, "Using the netlog," on page 113.	DSILOGP
NETVIEW.VnRnMn.*xxxxx*.DSILOGS	Contains the secondary network log. See Chapter 11, "Using the netlog," on page 113.	DSILOGS
NETVIEW.VnRnMn.*xxxxx*.DSITRCP	Contains the primary trace log.	DSITRCP
NETVIEW.VnRnMn.*xxxxx*.DSITRCS	Contains the secondary trace log.	DSITRCS
NETVIEW.VnRnMn.*xxxxx*.DSISVRT	Saves internal information for an eventual restart of msys for Operations.	DSISVRT
ING.*xxxxx*.STATS	Contains the automation status file. This file is used by the Sysplex Functions of msys for Operations.	AOFSTAT
ING.*xxxxx*.IPLDATA	Contains IPL information recorded after IPL. This file is used by the Sysplex Functions of msys for Operations.	HSAIPL

Note: The records needed for the system to view databases as active data sets are added during msys for Operations initialization.

Edit the sample job as described in the comments of the sample, and submit the job.

Loading members of partitioned data sets

This section documents data set members that you must copy from the target libraries to system or user libraries.

Loading DSIPARM members

Initially, the shared MSOPS.USER.DSIPARM data set and the system–specific MSOPS.USER.*xxxxx*.DSIPARM data sets (where *xxxxx* is the domain name) are empty. When performing the administration steps documented in this manual, copy the members you need to change from NETVIEW.VnRnMn.DSIPARM or ING.SINGSAMP to MSOPS.USER.DSIPARM or MSOPS.USER.*xxxxx*.DSIPARM. By doing this, you maintain a copy of the original member as it was installed from the distribution tape.

The edited members that can or should be kept in the shared MSOPS.USER.DSIPARM include:

- The customization member AOFCUST. For further information refer to "Editing the customization member AOFCUST" on page 46.
- The initialization style sheet C*xx*STYLE. For further information refer to "Customizing the initialization style sheet" on page 45.
- The backup command authorization table CNMSBAK1 (to be copied from NETVIEW.VnRnMn.CNMSAMP). For further information refer to "Protecting immediate commands when CMDAUTH=SAF" on page 80.
- The specification member DSIDMNK for some system level parameters.

Adding Procedures to PROCLIB

Copy the following members of ING.SINGSAMP into a library that is part of the PROCLIB concatenation:

- INGNVAP0

  This member contains a sample startup procedure MSOAPROC for msys for Operations. Rename the member to MSOAPROC. Review the procedure, at least for the high level qualifiers. See "Modifying the msys for Operations startup procedure" on page 44.
- INGPIXCU, INGPHOM, INGPIPLC, and HSAPIPLC

  These procedures are used internally. Follow the customization instructions, if any, that are contained within these procedures.

  Note: INGPIXCU and INGPHOM make use of certain data sets for which the started task users that are associated with these procedures and with NVSS must have the appropriate authorizations. For details, see "Granting NVSS and the STC-user access to data sets" on page 90.

The member CNMSJM04 in NETVIEW.VnRnMn.CNMSAMP contains a sample print job for the network log (see Chapter 11, "Using the netlog," on page 113) and the trace log. Copy this member to SYS1.PROCLIB and rename it to CNMPRT. If you defined passwords for the network log and the trace log (see "Defining passwords for VSAM databases" on page 99), add a password statement to CNMPRT.

Setting up the system logger

Because the HealthChecker backend stores all check results in the system logger, the following must be fulfilled:

- for stand-alone systems, **PLEXCFG=MONOPLEX** must be defined in SYS1.PARMLIB(IEASYS*xx*).
- systems in a sysplex must run in XCF mode and **PLEXCFG=MULTISYSTEM** 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 INGJFCDS member of the sample library. To define the log stream in:

- a single system environment, use the sample JCL provided in member INGJDLGM.
- a sysplex, use the sample JCL provided in member INGJDLGS.

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 according to the values below:

- Log stream definitions for CF

```
DATA TYPE(LOGR)

  DEFINE STRUCTURE NAME(ING_HEALTHCHKLOG)
         LOGSNUM(1)
         MAXBUFSIZE(65532)
         AVGBUFSIZE(1024)

  DEFINE LOGSTREAM NAME(ING.HEALTH.CHECKER.HISTORY)
         DESCRIPTION(HEALTH_CHECK_LOG)
         STRUCTNAME(ING_HEALTHCHKLOG)
         STG_DUPLEX(NO)
         LS_DATACLAS(NO_LS_DATACLAS)
         LS_MGMTCLAS(NO_LS_MGMTCLAS)
         LS_STORCLAS(NO_LS_STORCLAS)
         LS_SIZE(4096)
         AUTODELETE(YES)
         RETPD(365)
         HLQ(NO_HLQ)
         HIGHOFFLOAD(80)
         LOWOFFLOAD(0)
         DIAG(YES)
```

- Log stream for DASD only

```
DATA TYPE(LOGR)

  DEFINE LOGSTREAM NAME(ING.HEALTH.CHECKER.HISTORY)
         DESCRIPTION(HEALTH_CHECK_LOG)
         DASDONLY(YES)
         MAXBUFSIZE(65532)
         STG_DATACLAS(NO_STG_DATACLAS)
         STG_MGMTCLAS(NO_STG_MGMTCLAS)
         STG_STORCLAS(NO_STG_STORCLAS)
         STG_SIZE(4096)
         LS_DATACLAS(NO_LS_DATACLAS)
         LS_MGMTCLAS(NO_LS_MGMTCLAS)
         LS_STORCLAS(NO_LS_STORCLAS)
         LS_SIZE(4096)
         AUTODELETE(YES)
         RETPD(365)
```

```
                HLQ(NO_HLQ)
                HIGHOFFLOAD(80)
                LOWOFFLOAD(0)
                DIAG(YES)
```

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

```
  STRUCTURE NAME(ING_HEALTHCHKLOG)
         SIZE(8M)
         PREFLIST(cfname,cfname)
```

In this CFRM policy, you have to adapt the PREFLIST for structure ING_HEALTHCHKLOG. 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 privileged and/or trusted RACF status, refer to the chapter *"Planning for System Logger Applications;"* in *z/OS MVS Setting Up a Sysplex* for more information on how to define authorization to system logger resources. The names of the system logger resource used by msys for Operations are ING.HEALTH.CHECKER.HISTORY. All NetViews need to be authorized for accessing the log streams. They need update access for RESOURCE(ING.HEALTH.CHECKER.HISTORY) CLASS(LOGSTRM). For further information see the section *"Authorization for System Logger Application Programs"* in *z/OS MVS Setting Up a Sysplex*.

Preparing VTAM

Because msys for Operations is a VTAM application, you must define it to VTAM. msys for Operations supplies a sample definition for an application major node in the member INGVTAM of ING.SINGSAMP. Copy this member to MSOPS.USER.*xxxxx*.VTAMLST, which was allocated by INGALLC0, and edit it if needed. When you use MSO&SYSCLONE as your domain name, no changes are required.

Note: If you are using VTAM V4R4 or above, ensure that XCF signalling connectivity is in place in your sysplex, either with CTCs or with CF signalling structures.

Add this member to your VTAM configuration list ATCCON*xx*. A sample list is contained in member CNMS0003 of NETVIEW.VnRnMn.CNMSAMP. You can copy this sample to MSOPS.USER.*xxxxx*.VTAMLST and edit it as required.

Modifying the application (APPL) major node

You may want to modify the following operands of the major node definition:

- Password
- Domain ID
- Logmode table

Changing the password (PRTCT parameter)

The original password on the ACBpassword keyword in CNMSTPWD (%INCLUDEd in the initialization style sheet) is MSO&SYSCLONE. If you change this password, change *every* occurrence of PRTCT in INGVTAM to the same value. For example, if you change the password to PW006, then change:

```
MSO&SYSCLONE. APPL AUTH=(NVPACE,ACQ,PASS),PRTCT=MSO&SYSCLONE.,             X
```

to

```
MSO&SYSCLONE. APPL AUTH=(NVPACE,ACQ,PASS),PRTCT=PW006,EAS=4,             X
```

Changing the domain ID

When you change the domain ID from MSO&SYSCLONE to something else, you must change *every* occurrence of MSO&SYSCLONE in INGVTAM to the current domain ID, except for the password. For example, if you changed the domain ID to MYDOM, then you must also change the model definition

```
MSO&SYSCLONE.* APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=MSO&SYSCLONE.,            X
```

to

```
MYDOM* APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=MSO&SYSCLONE.,            X
```

Notes:

1. An APPL name *pre*fixed with MSO&SYSCLONE cannot have its *suffix* changed. The new name must retain the LUC suffix. Thus, MSO&SYSCLONE.LUC would have to be changed to MYDOMLUC.
2. If you code the optional ACBNAME operand on the APPL statement, it must match the APPL name in column 1. Therefore, if you change the domain ID, you must also change the value of ACBNAME.

Changing the logmode table (MODETAB parameter)

The member CNMS0001 of NETVIEW.VnRnMn.CNMSAMP contains a sample logmode table named AMODETAB that includes logmode entries for msys for Operations sessions. The application definitions in INGVTAM point to entries in this table with their MODETAB and DLOGMOD parameters, for example:

```
MODETAB=AMODETAB,DLOGMOD=DSILGMOD
```

When you want to use this table, copy it to MSOPS.USER.*xxxxx*.VTAMLST (where *xxxxx* is the domain name) and edit it as required. Subsequently, assemble and link-edit it with Job CNMSJ006 (contained in NETVIEW.VnRnMn.CNMSAMP.CNMSJ006). CNMSJ006 places the assembled table into MSOPS.USER.VTAMLIB.

The definitions of AMODETAB take effect the next time you start VTAM or issue the VTAM command

```
MODIFY NET,TABLE,NEWTAB=AMODETAB,OPTION=LOAD
```

If you already have a table that contains all the required entries change the value of the `MODETAB` parameter to your table name, if necessary.

Preparing the startup procedure and the initialization style sheet

When you start msys for Operations, you use the START procedure MSOAPROC (INGNVAP0); you must review that procedure. Furthermore, you must customize the style sheet that contains the initialization information for msys for Operations.

Modifying the msys for Operations startup procedure

MSOAPROC (INGNVAP0) was copied to the PROCLIB when you loaded partitioned data sets during installation (see “Adding Procedures to PROCLIB” on page 41). Make the following changes to the msys for Operations startup procedure (MSOAPROC).

Setting symbols

You can set the following JCL symbols in MSOAPROC:

Q1
This symbol determines the high level qualifier for user data sets. In the sample procedure, it is set to `ING.USER`.

DOMAIN
With this symbol you can specify the domain name of the msys for Operations instance to be started. The domain must be defined to VTAM with this name. The name must be unique across the network. It must not contain more than five characters.

You have three options for specifying the domain name:

1. In the initialization style sheet
2. In the startup procedure
3. In the START command

The START command has the highest priority, the style sheet the lowest.

In the sample procedure, the DOMAIN symbol is set to `MSO&SYSCLONE</xph>, where &SYSCLONE is a predefined system symbol. The default value of &SYSCLONE consists of the last two characters of the system name. However, you can override this value in IEASYM`*`xx`*`.`

You can accept the settings of MSOAPROC, and share the startup procedure among all systems of the sysplex, if the following conditions are satisfied:

- The value of &SYSCLONE is different for every system of the sysplex.
- You do not run an msys for Operations instance and a licensed NetView on the same system.

SQ1
This symbol determines the high level qualifier for the NVSS data sets. In the sample procedure, it is set to `NETVIEW.VnRnMn`.

SQ2
This symbol determines the high level qualifier for the Sysplex Functions data sets. In the sample procedure, it is set to `ING`.

VQ1
This symbol determines the high level qualifier for the VSAM clusters of NVSS. In the sample procedure, it is set to `NETVIEW.VnRnMn`.

NV2I
This symbol determines the name of the style sheet that is used for initialization of msys for Operations. It can also be used to construct unique names. The value must consist of exactly two characters. The default is 'NM'. In the sample procedure, no value is set.

The style sheet that msys for Operations is to use for initialization must have the name C*xx*STYLE, where *xx* is the value of &NV2I. For example, if you set &NV2I to N1, msys for Operations will expect to find a member CN1STYLE in DSIPARM. If you do not specify a value for &NV2I, the member CNMSTYLE will be used.

If you wish to use the same style sheet for all msys for Operations instances of the sysplex, do not specify a value for &NV2I and keep CNMSTYLE in MSOPS.USER.DSIPARM.

Customizing the initialization style sheet

When msys for Operations is started it reads a member of DSIPARM that contains the initialization parameters. This initialization member is called *style sheet*.

The name of the style sheet that msys for Operations will access is controlled by the value of &NV2I in the msys for Operations start procedure. The value of &NV2I replaces the second and third characters of the default name C**NM**STYLE. For example, a value of **E1** for &NV2I causes msys for Operations initialization to use member C**E1**STYLE in DSIPARM. For more information on NV2I, see "Setting symbols" on page 44.

For initialization of msys for Operations, you must customize the `SSIname` entry and the `TOWER` entry.

Defining the name of the automation router task

The automation router task receives the unsolicited MVS messages that are defined as automatable in MPFLST*xx*, and initiates the automated response if the automation table contains an entry for that message. You must specify the name of this task as the value of the `SSIname` keyword. It is essential that the name of this task is unique sysplex wide. This has the following reason:

The automation router task needs an EMCS console to receive MVS messages. The name of this console is identical with the task name. But EMCS consoles must have unique names across the sysplex. Therefore, the automation router task of every msys for Operations instance must have a different name from all other automation router tasks within the sysplex.

Provided that you do not run msys for Operations and a licensed NetView on the same system, you can ensure that the names are unique, and at the same time share the style sheet among the members of the sysplex, by defining

```
SSIname=&DOMAIN.SIR
```

where `&DOMAIN` has a different value on every system (for example, by setting the DOMAIN symbol to `MSO&SYSCLONE`, see "Domain names" on page 34). Keep the shared style sheet in MSOPS.USER.DSIPARM.

Note: msys for Operations does not use the subsystem interface (SSI) of NetView, but the automation router task must be active nevertheless.

Activating the SA tower

In order to enable msys for Operations for the Sysplex Functions, remove the asterisk before 'SA' in the TOWER statement, that is, change:

```
TOWER = *SA *AON  *MSM  *Graphics *AMI MVScmdMgt
```

to

```
TOWER = SA *AON  *MSM  *Graphics *AMI MVScmdMgt
```

Editing the customization member AOFCUST

AOFCUST is the customization member for the sysplex-related recovery actions of msys for Operations. It:

- Tells msys for Operations which recovery actions are to be performed automatically.
- Supplies information that msys for Operations needs for some of these recovery actions.

Note: By default, all recovery actions are disabled.

A sample of the AOFCUST member is contained in the ING.SINGNPRM library. It is recommended to edit a copy of this sample and copy it to the data set &*Q1*..DSIPARM specified in the startup procedure MSOAPROC. It is highly recommended that this data set is shared in the sysplex so that all instances of msys for Operations within the sysplex are customized identically.

AOFCUST consists of several sections.

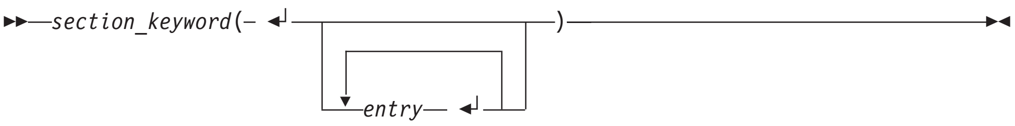
- "AUTO section – switching functions ON and OFF"
- "COMMON section – common definitions" on page 49
- "PAGE section – predefined and dynamically allocated local page data sets" on page 50
- "ENQ section – resources to be monitored and jobs to be canceled" on page 53
- "HEALTHCHK section – specifying user overrides" on page 57
- "CDS section – spare volumes for CDS recovery" on page 69
- "HW section – hardware configuration" on page 70
- "IXC102A section – hardware commands" on page 71
- "WTOBUF section – jobs to be canceled in case of buffer shortage" on page 73

The format of these sections is described by syntax diagrams. Note the following convention for these diagrams:

Line break symbol in the syntax diagrams

In the syntax diagrams IN this chapter, the symbol ↵ denotes a line break.

The format of a section is:



Every entry must be placed in a separate line. The closing parenthesis of a section must also be on a separate line. The maximum length of any line is 72 characters.

The entries can consist of only one keyword, but they can also have an internal structure of their own. You can comment out an entry by entering an asterisk in column 1.

For checking the syntactical correctness of AOFCUST, msys for Operations provides the INGCUST command. For details, see "INGCUST" on page 197.

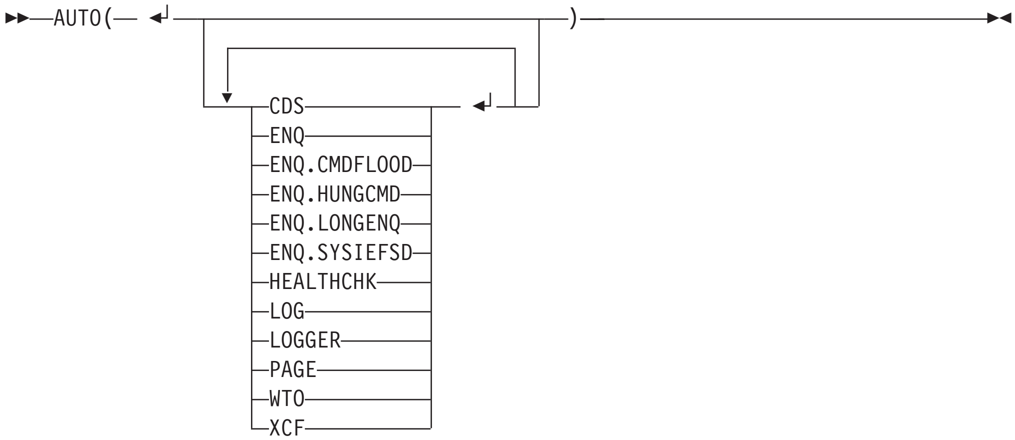
AUTO section – switching functions ON and OFF

Purpose

You can enable or disable the recovery functions of msys for Operations in the `AUTO` section. By default, all recovery actions are disabled.

Format

AUTO section – switching functions On and Off



Parameters

CDS Switches on background recovery of alternate CDSs. Automatic recreation of CDSs with INGPLEX CDS is always enabled. It cannot be switched off in the `AUTO` section. For general information, see “Ensuring continuous availability of alternate couple data sets” on page 9.

For recovery of alternate CDSs, you must specify the spare volumes on which msys for Operations may allocate the new alternate CDSs in the `CDS` section.

ENQ Enables the handling of the next four individual recoveries.

ENQ.CMDFLOOD
Enables the handling of commands that flood a particular command class.

ENQ.HUNGCMD
Enables the handling of jobs and commands that inhibit other commands from completing execution.

ENQ.LONGENQ
Enables the handling of long-running ENQs.

ENQ.SYSIEFSD
Enables the handling of ENQs related to the major resource SYSIEFSD and the minor resources Q4 and Q10.

HEALTHCHK
Enables the HealthChecker, which checks the current, active sysplex settings and definitions for an image, and compares these values to those either suggested by IBM (known as *best practices*) or defined by you.

LOG Switches on recovery of the system log. For general information, see “Resolving a system log failure” on page 12.

LOGGER
Switches on the recovery actions relating to the system logger. For general information, see “Managing the system logger” on page 10.

Note that if you switch on automation for system logger recovery, you also activate the recovery function for alternate CDSs, even if CDS automation (`CDS` keyword) is switched off. Therefore, if you decide to use system logger automation, but do not want msys for Operations to recreate alternate CDSs for some or all CDS types, you must delete any spare volumes defined for the types in question.

PAGE Prevents auxiliary storage shortages by predefining local page data sets.

WTO Switches on recovery of WTO(R) buffer shortages. For general information, see "Resolving WTO(R) buffer shortages" on page 12.

For WTO(R) buffer recovery, specify the jobs that may be canceled in order to resolve the buffer shortage in the `WTOBUF` section.

XCF Switches on the automation of the IXC102A and IXC402D messages. If XCF is enabled, make sure that the following conditions are met:

- msys for Operations must run on all systems in the sysplex because the message is issued only on one system
- The failing system must be enabled for this automation
- The BCP Internal Interface must be available
- The failing system must be running on a G3 or follow-on CPC

Because the automation must know where the system is located, to send the hardware command to the appropriate Support Element you must predefine your hardware configuration in the `HW` section of AOFCUST.

Furthermore, if you want to define hardware commands other than SYSRESET, or to disable the automation for some systems, you need to supply definitions in the `IXC102A` section of AOFCUST. Note that only the following hardware commands are accepted:

- SYSRESET
- LOAD
- ACTIVATE
- DEACTIVATE

Example

In the following example, the first system is re-IPLed when the automation takes place. The second system is deactivated, and the third system is not automated at all. All systems not specified are automated depending on the flag XCF in the AUTO section. For these systems the default action SYSRESET CLEAR is being performed.

```
IXC102A(
  CMD(sys1,LOAD CLEAR)
  CMD(sys2,DEACTIVATE)
  DISABLE(sys3)
)
```

COMMON section – common definitions

Purpose

This section defines values that are common to all functions of msys for Operations. Note that the TEMPHLQ parameter is required by the automation.

Format

```
►►──COMMON(── ◄┘──TEMPHLQ──(──hlq──)── ◄┘─────────────────────────────── ◄┘──)────►◄
                                         └─STCJOBNM──(──jobname──)─┘
```

Parameters

TEMPHLQ
This keyword introduces the high-level qualifier which is used to assemble a data set name for allocating temporary data sets needed by programs running as started tasks.

hlq The high-level qualifier can consist of 17 characters. The name must comply to the MVS naming rules.

STCJOBNM
This keyword introduces the job name being used for programs running as started tasks. When not defined the job name of each started task defaults to the procedure name.

jobname
The job name may consist of up to 8 characters. The name must comply to the MVS job naming rules.

Example

```
COMMON(
  TEMPHLQ (AOC.TEMP)
  STCJOBNM(AOCSTC)
)
```

PAGE section – predefined and dynamically allocated local page data sets

Purpose

The recovery prevents outages caused by auxiliary storage shortages.

To enable the function add the PAGE keyword to the existing section AUTO in the member AOFCUST. Note that also the PAGTOTL parameter defined in one of IEASYS*xx* PARMLIB members used during the IPL must specify a value greater than the number of local page data sets currently used. And, you must have defined at least one predefined local page data set or a spare volume including the high level qualifier.

Format

```
►►──PAGE(─────────────────────────────────────────────────────────────►

►─ ◄┘─┬──────────────────────┬──┬───────────────────────────────────┬─►
       └┤ Predefined Data Set ├┘  └┤ Dynamically Allocated Data Set ├┘

►─┬───────┬─ ◄┘─)─────────────────────────────────────────────────────►◄
  └┤ Job ├┘
```

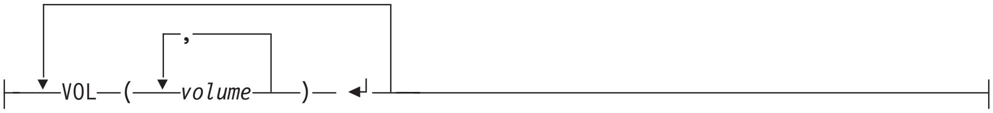
Predefined Data Set

```
   ┌────────────────────────┐
   ▼                        │
├──DSN──(──dsname──)── ◄┘───┴──────────────────────────────────────────┤
```

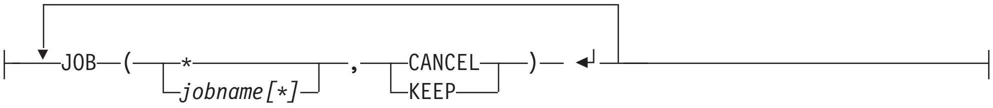
Dynamically Allocated Data Set

```
├──HLQ──(──hlq──)── ◄┘──CYL──(──nnn──)── ◄┘──┤ Volume List ├──────────┤
```

Volume List



Job



Parameters

CANCEL
This keyword indicates that the specified job is canceled when the job caused the shortage and the shortage cannot be relieved within a specific time frame.

CYL This keyword defines the space being used on dynamic allocations of page data sets.

DSN This keyword defines the data set name of a predefined spare local page data set.

HLQ This keyword defines the high level qualifier of dynamically allocated page data sets.

JOB This keyword defines the job being checked when shortage condition occurs.

KEEP This keyword indicates that the specified job must not be canceled when the job caused the shortage.

VOL This keyword defines a list of volumes where a local page data set can be allocated dynamically.

dsn Is the name of a predefined local page data set. The data set must be allocated on a volume shared by all systems in the sysplex and catalogued in the master catalog. In case the systems in the sysplex do not share the master catalog the page data set must be recatalogued on each system except the one which created the data set using the following IDCAMS statement:

```
DEF PAGESPACE(NAME(dsn) VOLUME(volume) RECATALOG)
```

hlq Is the high-level qualifier which is used to assemble a data set name for creating and allocating a page data set. The qualifier may consist of up to 23(!) characters and must comply to the MVS data set naming rules.

Note: The high level qualifier must point to the master catalog and must not be SMS managed.

jobname
Is the name of a job or a name pattern. Specify KEEP or CANCEL for jobs that can or cannot be canceled. You can use wildcards when specifying the job name.

nnn Is the space in cylinders (100-999) being used for the dynamic allocation of a local page data set.

volume Is the serial number of a volume where a local page data set can be allocated.

Note: The volumes must be shared by all systems in the sysplex.

Usage

To customize the automation, you can define the following:

- The predefined local page data sets that can be immediately added via PAGEADD.
- The potential volumes where new local page data sets can be allocated.
- The high-level qualifiers of those local page data sets being dynamically allocated. The actual data set name is built by concatenating the qualifier by ′.V′ followed by the volume serial number being used in the allocation.
- The amount of space for each dynamic allocation of a local page data set. If omitted a default value of 400 is used.

  Note: On a 3390 DASD 100 cylinders are adequate to 70 MB. The formatting process lasts approximately 18 seconds for this amount of space.
- The names of jobs that can or cannot be canceled. If no ″JOB(*,...)″ statement is defined, the following statement is generated:

  ```
  JOB(*,KEEP)
  ```

All keywords except HLQ and CYL can be specified as often as needed. Up to 8 volume names can be specified per VOL statement. When both JOB(*,CANCEL) and JOB(*,KEEP) are specified the KEEP statement takes precedence.

The qualifiers of the HLQ statement may contain system symbols like &SYSNAME. If a symbol cannot be substituted it is replaced by its name. The qualifier is appended by the system name followed by ′.Vvvvvvv.Snn′ to assemble the data set name where ′v′ is the volume serial number and ′n′ is a sequence number from 00 to 99.

The recommendation is to use predefined spare local page data sets rather than dynamically allocated data sets. This minimizes the time of the storage shortage. Each predefined local page data set should be allocated with 10% space of local page space currently used by the system.

Note: After a predefined spare local page data set has been made available by the automation to any system in the sysplex and it is no longer being used, you must delete the data set manually using the PAGEDEL to return the data set to pool of spare local page data sets.

Example

```
PAGE(
  DSN (SPARE.LOCAL.PAGE.DS1)
  DSN (SPARE.LOCAL.PAGE.DS2)
  HLQ (SPARE.DYNLOCAL.PAGE)
  CYL (600)
  VOL (vol111,vol222,vol333)
  VOL (vol999)
  JOB (BTCH*,CANCEL)
  JOB (*,KEEP)
)
```

ENQ section – resources to be monitored and jobs to be canceled

Purpose

This section defines the resources that are to be monitored or excluded from monitoring when you want to automate long running ENQs, ″hung″ commands and command flooding.

As soon as one exclude statement is defined for either long-running ENQ recovery (RES) or ″hung″ command recovery (CMD), the particular processing is treated as exclusion processing.

Note: No customization is possible for the recovery of the SYSIEFSD family of minor resources (Q10 and Q4). The time interval for these resources it is 10 seconds.

You must define the following if you have enabled long-running ENQ recovery:

- The resource (or resources) being monitored or being excluded from monitoring
- The time frame for each resource when an ENQ on the resource is treated as a long running ENQ or, if resources are to be excluded from monitoring, the exclusion keyword

You must define the following if you have enabled ″hung″ command recovery:

- The command (or commands) being monitored or being excluded from monitoring
- The time frame for each command that a command is granted to finish or, if commands are to be excluded from monitoring, the exclusion keyword

To prevent an outage you may define what jobs that are locking the resource can be canceled when a long-running ENQ, a ″hung″ command, or command flooding is detected. You also can define if those jobs are being dumped before they are canceled. You can define:

- The names of the jobs that should be canceled or kept when originating a long-running ENQ, causing a ″hung″ command situation, or flooding a command class
- The snapshot interval for a command class
- The title of the dump taken before a job is canceled
- The default storage areas to be dumped
- The symbol definitions to be used when the dump specifications are provided by a PARMLIB member

Format

```
►►──ENQ(──┬──────────────────┬──┬─────────────────┬──┬─────────────┬──────────────►
          └─◄┘─┤ Resource ├─┘  └─◄┘─┤ Command ├─┘  └─◄┘─┤ Job ├─┘

►──┬────────────────────────┬──┬───────────────────────────┬──────────────────────►
   └─◄┘─DUMP──(─sdata─)─┘  └─◄┘─┤ Symbol Definition ├─┘

►──┬──────────────────────────┬──┬──────────────────────────┬── ◄┘─)──────────►◄
   └─◄┘─TITLE──(─title─)── ◄┘┘  └─◄┘─┤ Command Flooding ├─┘
```

ENQ section – resources to be monitored and jobs to be canceled

Resource

```
   ┌───────────────────────────────────────────────────┐
   ▼                                                   │
├──RES──(──qname──,──rname──,──┬─waittime─┬──)── ◄┘────┴──────────────────┤
                               └─Exclude──┘
```

Command

```
   ┌─────────────────────────────────────────────────────────────────────┐
   ▼                                                                     │
├──CMD──(──command──[──parms──]──,──rname──,──┬─waittime─┬──)── ◄┘───────┴──────┤
                                              └─Exclude──┘
```

Job

```
   ┌─────────────────────────────────────────────────────┐
   ▼                                                     │
├──JOB──(──┬─*───────────┬──,──┬─DUMP─────┬──)── ◄┘──────┴───────────────────┤
           ├─jobname[*]──┤     ├─KEEP─────┤
           └─asid────────┘     ├─NODUMP───┤
                               └─xx[,xx]──┘
```

Symbol Definition

```
   ┌───────────────────────────────────────────────────────────────────┐
   ▼                                                                   │
├──SYMDEF──(──┬─*───────────┬──,──[&]&symbol.──=──'value'. ──)── ◄┘────┴───────┤
              ├─jobname[*]──┤
              └─asid────────┘
```

Command Flooding

```
   ┌─────────────────────────────────────────────────┐
   ▼                                                 │
├──FLOOD──(──class──,──snapshottime──)── ◄┘──────────┴─────────────────────────┤
```

Parameters

RES This keyword defines the resource to be monitored or excluded from monitoring.

Exclude
This keyword defines the resource (or resources) that are to be excluded from monitoring.

Specifying one ′Exclude′ statement requires the generic definition of

```
RES(*,*,waittime)
```

to complete the exclusion list.

CMD This keyword defines the commands to be monitored or excluded from monitoring.

JOB This keyword defines the job being checked when a long running ENQ is detected.

DUMP
This keyword defines the dump options being used when a dump is taken before the job is canceled.

On the JOB statement it indicates that the specified job is canceled with a dump using the default dump options when the job is the owner of a long running ENQ.

KEEP This keyword indicates that the specified job must not be canceled when the job is the owner of a long running ENQ.

NODUMP
This keyword indicates that the specified job is canceled without a dump when the job is the owner of a long running ENQ.

SYMDEF
This keyword defines the symbol being substituted when a dump is taken.

TITLE This keyword defines the default dump title when a dump is taken.

FLOOD
This keyword defines the command class that you want to perform recovery for.

qname is the major resource name being checked. You can use wildcards when specifying the major resource name.

rname is the minor resource name being checked. You can use wildcards when specifying the minor resource name.

waittime
is the time in seconds (30 — 999) after the automation treats an ENQ as a long blocking ENQ.

command
is the name of the MVS system command being checked. You can use wildcards when specifying the command name.

parms specifies the command parameter (or parameters) of the MVS command being checked. You can use wildcards when specifying the parameter.

jobname
is the name of a job or a name pattern. You can use wildcards when specifying the job name.

asid is the four-character address space ID, for example, 000A.

xx is the suffix of a IEADMC*xx* PARMLIB member describing the dump specifications. This suffix is used on the dump command instead of the default dump options when a job is canceled due a long running ENQ detection.

sdata specifies the storage areas to be dumped when the keyword DUMP is used instead of a PARMLIB member. If omitted the following areas are assumed:

CSA	Common Service Area
GRSQ	Global resource serialization (ENQ/DEQ/RESERVE) queues
RGN	Private area of address space being dumped, including LSQA (Local System Queue Area) and SWA (Scheduler Work Area)
SQA	System queue area

NOSUM	No summary dump
TRT	GTF, system trace, master trace, and NIP hardcopy buffer data

For further details of dump options refer to the description of the MVS DUMP command.

symbol is the name of a system symbol to be substituted.

value is the substitution value.

title is the title of the dump. The title can be up to 100 characters in length. If omitted the title defaults to *"Dump by msys for Operations due to a long ENQ detection"*.

class is one of the command classes: M1/M2/M3/C1/C2/C3.

snapshottime
is the time in seconds (1 - 60) that the recovery routine should wait before it takes a snapshot of the commands that are being executed or waiting in the command class.

The time value should be long enough to allow a job just to finish the issuing of a set of commands like "VARY dev" in this time frame. This prevents such a job from unnecessarily being evaluated by command flooding recovery.

The default value is 3 seconds.

Restrictions

Defining any resource to be excluded from monitoring requires the generic resource definition

```
RES(*,*,waittime)
```

If you omit this definition, the automation assumes the following default:

```
RES(*,*,30)
```

Usage

The syntax rules for the PARMLIB suffix(es) and the symbol definitions comply to the rules of the MVS DUMP command. For details see *MVS System Commands*.

The keywords RES, JOB, and SYMDEF can be specified as needed. Specifying KEEP on the JOB statement means this job must not be canceled regardless of the locks being held. The remaining specifications DUMP, NODUMP, and the suffix(es) allow the cancellation of the job after the wait time has expired. NODUMP suppresses the dump before the job is canceled.

Note that to use ENQ automation, GRS must be active and the software prerequisites must be fulfilled, see "Software prerequisites" on page 29.

Example

```
ENQ(
  DUMP  (CSA,RGN,SUM,GRSQ)
  JOB   (0001,KEEP)
  JOB   (CICS*,KEEP)
  JOB   (ABC*,NODUMP)
  JOB   (BTCH*,D0,D1)
  JOB   (*,DUMP)
  RES   (MAJOR1,MINOR1,30)
  RES   (MAJOR2,*,999)
  TITLE (Dump by automation due to a long ENQ detection)
)
```

HEALTHCHK section – specifying user overrides

Purpose

In this section, you can override IBM's suggestions, which are hard coded.You can either:

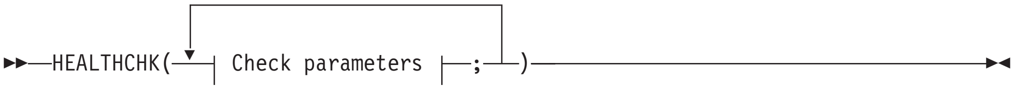
- Enable or disable the HealthChecker function as a whole—this is done in the AUTO section of AOFCUST:
  - – If you enable the function in AOFCUST, you can disable it temporarily for a certain period of time with the INGAUTO command.
  - – Conversely, if you disable the function in AOFCUST, you can enable it temporarily for a certain period of time with the INGAUTO command.
  - – While the HealthChecker is enabled, checks are repeated, based on individual time intervals that have been set for each check.
- Specify your own values for a check
- Disable the running of a check

The syntax requirements are described in "Format," and also in the AOFCUST member, see Figure 3 on page 61. You must provide both a DATE and REASON parameter in your user overrides:

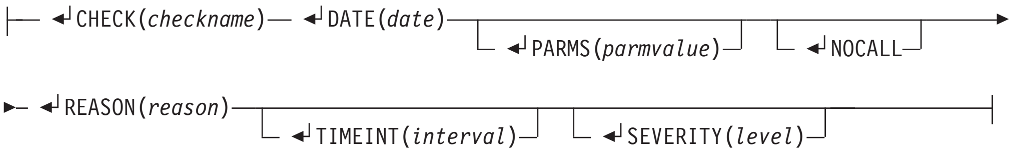
- The DATE is entered in the format *yyyymmdd*.
- The REASON is used as your comment area to describe your rationale for the override. These comments can be up to 100 characters long. Your comments (REASONs) are also displayed in the report output.

If you do not want to specify any user preferences, you can leave the HEALTHCHK section empty.

Format



Check parameters:



Parameters

CHECK
: This keyword indicates the check that is to be overridden.

DATE This keyword defines the date when the override was made.

PARMS
: This keyword defines the parameters, if applicable, for the check.

NOCALL
This keyword indicates that the check will be suppressed.

REASON
This keyword defines the reason for the override. It can be a maximum of 100 characters long, and is enclosed in single quotes.

TIMEINT
This keyword overrides the predefined time interval in which the check is repeated.

SEVERITY
This keyword overrides the predefined severity used to flag report data for the check.

checkname
This variable is the name of the check. See Table 17 on page 331 for a list of the checks and an indication of whether they are local or global, and what their interval is.

date This is the date that the user last defined the override. It is in *yyyymmdd* format and must be later than the corresponding date in the IBM best practices.

parmvalue
This variable indicates the user-defined values for the check. More detailed information about specific parameters can be found in the AOFCUST member (see Figure 3 on page 61).

reason This variable is the text string of the REASON keyword. This is the reason why you want to perform the override.

interval
This variable is the time interval, in hours and minutes, after which the check is repeated, in the format *hh:mm*.

level This variable is the severity used to flag report data for the check. It can be High (H), Medium (M), or Low (L).

Usage

Although the checks and IBM best practices values that are used by the HealthChecker are suitable for most enterprises, you may decide that you want to customize the HealthChecker to better meet your needs. You can override IBM's recommendations in the AOFCUST data set by:

- changing values
- suppressing IBM checks

You should create a copy of AOFCUST to make your updates. This ensures that your preferences will not be overwritten if a new level of the HealthChecker is installed. The syntax requirements are described in the AOFCUST member, as shown in Figure 3 on page 61.

Rules: The following rules apply for the user policy:

- The USER reason appears in the report if there is an exception to the user override.
- *Current design rationale for the DATE parameter:* Most users will use the IBM defaults, however there are some that have unique configurations (as they document in AOFCUST). If and when IBM changes the default values, we would like those of you that specified your own values to re-examine the rationale behind the values that you have specified. We can accomplish this by

updating the IBM date when the IBM defaults are changed, and then enforcing the rule that the USER dates must be newer than the IBM date.

As suggested in “Usage” on page 58 above, you should create a copy of AOFCUST to make your updates. Thus, your preferences will not be overwritten when a new level of the HealthChecker is installed. You must then evaluate the new AOFCUST, comparing its dates to those in your prior level of AOFCUST. Any desired overrides must have a date later than that shipped in the new level of AOFCUST. This ensures that you will get the expected results. If an IBM date is more current than the DATE for the override of the parameter, then the HealthChecker will issue an error message.

We enforce a later date in AOFCUST to ensure that customers re-evaluate their decision to override IBM best practices after IBM makes a change to them. If the date on these is later than the date in the AOFCUST, the HealthChecker will not perform the check. Thus, we force either an update of the date in AOFCUST, or the customer decides not to override IBM best practices any longer.

Changing parameter values: You can override any IBM check that uses the PARM statement to specify the values to be used. The report will display the results of the checks, the values used to evaluate the check, and the rationale of the check. A number of messages also provide actions and references to additional information. The IBM checks, including those that can be overridden, are described in Appendix E, “The IBM Health Checker for z/OS and Sysplex checks,” on page 331.

Example: This example shows how you can modify the check for the number of EMCS consoles. The original check that uses IBM’s best practices is as follows:

```
CHECK(Number_EMCS_consoles)
        DATE(20030102)
        PARMS(5000,10000)
        REASON('Excessive numbers of EMCS consoles cause slowdown');
```

This check verifies that the number of active EMCS consoles is less than 5000, and that the number of inactive EMCS consoles is less than 10000.

The following example shows how to specify your own, preferred values for the number of active and inactive EMCS consoles. Suppose that you update AOFCUST, specifying that the check should be for more than 25 active EMCS consoles and more than 100 inactive EMCS consoles. The text in **bold** indicates the values that you must change.

```
CHECK(Number_EMCS_consoles)
        DATE(20030401)
        PARMS(25,100)
        REASON('Reduced IBM values of 5000, 10000 to 25,100.');
```

The TIMEINT and SEVERITY parameters can be changed in a similar way.

Suppressing an IBM check: There may be some checks that you don’t feel are applicable for your environment. You can omit these checks from the HealthChecker. The following example shows how to specify that the check for number of EMCS consoles is not performed. To do this, use the NOCALL parameter.

To suppress this check, specify the following. The text in **bold** indicates the values that you must change.

```
CHECK(Verify_numbers_EMCS_consoles)
        DATE(20030401)
        NOCALL
        REASON('This check is not valid for our environment.');
```

Note: Running the system console in PD mode can be of value for customers. On the other hand, enabling this function affects the overall performance of the system. Thus the HealthChecker advises customers to turn off this function unless it is really needed. Customers who activate the automation of messages IXC102A or IXC402D, or both, need this function and should consider disabling the check (override it with NOCALL).

Note: Running the system console in PD mode can be of value for customers. On the other hand, enabling this function affects the overall performance of the system. The HealthChecker hence advises customers to turn off this function, unless it is really needed. Customers who activate the automation of messages IXC102A or IXC402D, or both, have a need for this function and should consider disabling the check (overriding it with NOCALL).

Example

Figure 3 on page 61 shows an example of the HEALTHCHK section of AOFCUST.

```
* -------------------------------------------------------------------
* Health Checker section
* -------------------------------------------------------------------
* This is the definition of the user's best practices (USERPARM) for
* the Health Checker.
*
* In the following, IBMPARM means the IBM best practices as shown
* by command INGPLEX BESTpractices.
*
* USER INSTRUCTIONS:
* 1. Use USERPARM to override IBMPARM.
* 2. Follow documentation below for details of each check.
* 3. Specify PARMS, TIMEINT, SEVERITY, or NOCALL,
*    as well as REASON, and DATE.
*    DATE in USERPARM should be later than DATE in IBMPARM,
*    REASON in USERPARM should be different from REASON in IBMPARM.
*
* SYNTAX:
* 1. 1st keyword must be CHECK
* 2. End each statement with a semi-colon.
* 3. Use cols 1-72 only for statements.
* 4. Keywords on each statements are:
*    CHECK    - must match a CHECK in IBMPARM.
*    REASON   - max 100 chars - enclose in quotes. For formatting
*               reasons it is recommended that individual words not
*               span lines because a blank is inserted automatically
*               with each line break.
*               Internal processing reduces a sequence of blanks to one
*               blank, so for each such sequence reduce the allowed
*               maximum of 100 characters by 1.
*               If REASON spans multiple lines, then for each but the
*               last line subtract one inserted blank per line from the
*               allowed maximum of 100 characters.
*    DATE     - yyyymmdd - must be later than matching date in IBMPARM.
*               Since the value specified is used for a simple compare
*               with IBM's date, the input values are checked for
*               validity only as far as needed, i.e. not for total
*               match with the calendar, i.e. 20030231 would be
*               accepted.
*    SEVERITY - HIGH, MEDIUM, or LOW - overrides the predefined
*               severity used to flag report data for the check
*    TIMEINT  - hh:mm - overrides the predefined time interval in
*               which the check is repeated - in hours and minutes.
*    NOCALL   - eliminates a check.
*               If you specify both, NOCALL and PARMS, the check is
*               still eliminated
*    PARMS    - format varies with each check - see IBMPARM.
**-------------------------------------------------------------------
*
* The following are the definitions from IBMPARM, they can be used as
* samples to set USERPARM.
*
*HEALTHCHK(
*
* -------------------------------------------------------------------
*      CouplingFacility_Structure status:
* -------------------------------------------------------------------
*
*   Create a list of all Systems defined and report on their status.
*
*   Create a list of all CFs defined and report on their status.
*
*   Create a list of all STRuctures defined and report on status.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(CouplingFacility_Structure)
*       Severity(Medium)
*       DATE(20030102)
*       reason('Check CF and Structure location');
*
```

Figure 3. AOFCUST section HEALTHCHK (Part 1 of 8)

```
* -------------------------------------------------------------------
*      CouplingFacility_Structure descriptions:
* -------------------------------------------------------------------
*
*   Create a report showing what systems, CFs, and structures are
*   defined in the sysplex. This report will show appropriate status
*   and connection status of these resources.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(Sys_CF_STR_Report)
*       Severity(Low)
*       DATE(20030102)
*       reason('Create System, CF, Structure report');
*
* -------------------------------------------------------------------
*   XCF_Signalling checks:
* -------------------------------------------------------------------
*   Check #1:
*      Check that ALL transport classes are set up to service
*      the pseudo-group name 'UNDESIG ',
*      i.e. that any XCF message can use each transport class.
*   Check #2:
*      Check that all defined Transport Classes are assigned to
*      at least one pathout (outbound path).
*   Check #3:
*      Check that most pathouts have a transport class defined
*      with a "small" (parm4) classlength and at least one other
*      transport class is defined with a higher "large" (parm5)
*      classlength.
*   Check #4:
*      Check that multiple (at least parm3) Pathout/Pathin pairs
*      are in the WORKING (i.e. OPERATIONAL) state for each system
*      in the sysplex connected to the current system.
*   Check #5:
*      Check that there is a MAXMSG of at least the indicated
*      minimum value (parm1) for each transport class.
*   Check #6:
*      Check each inbound signal path and ensure that each can
*      support at least the indicated minimum number (parm2) of
*      messages from the sending system.
*      (AMDPMXMS / (AMDPATH1_BuffLen + 2K)) should be > specified
*      minimum number of messages supported by the path.
*
* PARAMETERS:
*   Check #1: None required.
*   Check #2: None required.
*   Check #3: Parameters 4 and 5
*   Check #4: Parameter 3
*   Check #5: Parameter 1
*   Check #6: Parameter 2
*
*   XCF_Signalling positional parameter descriptions:
*
*   1. Check #5 parameter 1:
*      The minimum MAXMSG value for transport classes.
*      This is an INTEGER. The maximum acceptable value is 999999.
*
*   2. Check #6 parameter 2:
*      The minimum number of XCF messages that an inbound
*      XCF signal path should support to avoid message backup.
*      This is an INTEGER. The maximum acceptable value is 999999.
*
```

Figure 3. AOFCUST section HEALTHCHK (Part 2 of 8)

```
*   3. Check #4 parameter 3:
*      Specifies the minimum pathout/pathin pair count
*      for a system.
*      This is an INTEGER. The maximum acceptable value is 9.
*
*   4. Check #3 parameter 4:
*      Specifies the maximum value to be interpreted as a "small"
*      (XCF transport) classlength.
*      This is an INTEGER. The maximum acceptable value is 9999.
*
*   5. Check #3 parameter 5:
*      Specifies the minimum value to be interpreted as a "large"
*      (XCF transport) classlength.
*      This is an INTEGER. The maximum acceptable value is 62464.
*                          The minimum acceptable value is 4028.
*      The specified value does not include the 68 additional bytes
*      used by XCF for internal control blocks
* ------------------------------------------------------------------
*CHECK(XCF_Signalling)
*       Severity(Medium)
*       DATE(20030102)
*       PARMS(750,30,2,956,4028)
*       REASON('Avoid problems with XCF signalling.');
*
* ------------------------------------------------------------------
*   XCF signalling structures in coupling facilities:
* ------------------------------------------------------------------
*   When XCF signalling structures in coupling facilities are used,
*   check that:
*     1. not all the signalling structures reside on the same
*        coupling facility (CF).
*     2. multiple links (or CHPIDs) to each CF are both
*        ONLINE and OPERATING.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(XCF_Signalling_Structures_in_CF)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Avoid problems with XCF signalling in CFs.');
*
* ------------------------------------------------------------------
*    CONSOLE Names:
* ------------------------------------------------------------------
* Check that each Console has the NAME parameter specified
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Console_Names)
*       Severity(High)
*       DATE(20030102)
*       REASON('Like named consoles are matched across the sysplex');
*
* ------------------------------------------------------------------
*  Alternate CONSOLE Groups:
* ------------------------------------------------------------------
* Check that each Console has the ALTGRP parameter specified.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Alternate_Console_groups)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Provides good recovery from console loss');
*
```

Figure 3. AOFCUST section HEALTHCHK (Part 3 of 8)

```
* ------------------------------------------------------------------
*  Master authority:
* ------------------------------------------------------------------
* Check that each system has a console with MASTER authority.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Console_master)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Needed for DCCF and other situations');
*
* ------------------------------------------------------------------
*   CONSOLE MSCOPE versus Routcodes:
* ------------------------------------------------------------------
* Check that each console has an acceptable mix of MSCOPE and
* Routcodes.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Console_MSCOPE_and_Routcodes)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Avoids overloading any console. Reduces number of
*messages sent to Sysplex consoles');
*
* ------------------------------------------------------------------
*  AMRF and Eventual Action message retention:
* ------------------------------------------------------------------
* If AMRF is ON, check that Eventual_Action messages are not
*    retained.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(AMRF_and_MPF_consistent)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Avoids long chains of messages in storage');
*
* ------------------------------------------------------------------
*  CONSOLEs and ROUTCODE 11:
* ------------------------------------------------------------------
* Check that no console is receiving ROUTCODE 11 messages.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Console_routcode_11)
*       Severity(Low)
*       DATE(20030102)
*       REASON('Not really needed as for programmer info only');
*
* ------------------------------------------------------------------
*  Sysplex Master Console:
* ------------------------------------------------------------------
* Check that the MASTER console is active.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Sysplex_master)
*       Severity(High)
*       DATE(20030102)
*       REASON('Needed in emergencies');
*
```

Figure 3. AOFCUST section HEALTHCHK (Part 4 of 8)

```
*  EMCS consoles and MSCOPE versus Routcodes:
* -----------------------------------------------------------------
* Check that each EMCS console has an acceptable mix of MSCOPE and
* Routcodes.
*
* PARAMETERS: None
* -----------------------------------------------------------------
*CHECK(EMCS_MSCOPE_and_Routcodes)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('ROUTCODE(ALL) and non-local MSCOPE will cause a large
*number of messages to be processed');
*
* -----------------------------------------------------------------
*  EMCS Consoles and HARDCOPY Flag:
* -----------------------------------------------------------------
* Check that each EMCS console does not have the HARDCOPY flag set
* if MSCOPE > 1.
*
* PARAMETERS: None
* -----------------------------------------------------------------
*CHECK(EMCS_hardcopy)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('EMCS consoles with HARDCOPY specified will process an
*excessive number of messages');
*
* -----------------------------------------------------------------
*  SYSCONS and MSCOPE:
* -----------------------------------------------------------------
* Check that the SYSCONS has only local MSCOPE.
*
* PARAMETERS: None
* -----------------------------------------------------------------
*CHECK(SYSCONS_MSCOPE)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('If SYSCONS is used in emergencies it should not have
*to process large numbers of messages');
*
* -----------------------------------------------------------------
*  SYSCONS and ROUTCODE:
* -----------------------------------------------------------------
* Check that the SYSCONS has advisable routcodes.
*
* PARAMETERS: None
* -----------------------------------------------------------------
*CHECK(SYSCONS_ROUTCODES)
*       Severity(Low)
*       DATE(20030102)
*       REASON('If SYSCONS is used in emergencies it should not have
*to process large numbers of messages');
*
* -----------------------------------------------------------------
*  Number of EMCS Consoles
* -----------------------------------------------------------------
* Check that there is not an excessive number of EMCS consoles.
*
* PARAMETERS:
*       1. Maximum number of ACTIVE EMCS consoles on this system.
*          Values between 0 and 99999999 are accepted.
*          Must be numeric.
*       2. Maximum number of INACTIVE EMCS consoles on the entire
*          sysplex. Values between 0 and 99999999 are accepted.
*          Must be numeric.
* -----------------------------------------------------------------
```

Figure 3. AOFCUST section HEALTHCHK (Part 5 of 8)

```
*CHECK(Number_EMCS_consoles)
*       Severity(High)
*       DATE(20030102)
*       PARMS(5000,10000)
*       REASON('Excessive numbers of EMCS consoles cause slowdown');
*
* ------------------------------------------------------------------
*  SYSCONS and PD mode:
* ------------------------------------------------------------------
* Check that SYSCONS is not in PD mode
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(SYSCONS_PD_MODE)
*       Severity(Low)
*       DATE(20030102)
*       REASON('SYSCONS should be run in Problem Determination mode
*only when there is a problem');
*
* ------------------------------------------------------------------
*  SYSCONS and MASTER authority:
* ------------------------------------------------------------------
* Check that SYSCONS has MASTER authority
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(SYSCONS_MASTER)
*       Severity(High)
*       DATE(20030102)
*       REASON('SYSCONS needs MASTER authority to resolve problems in
*emergency situations');
*
* ------------------------------------------------------------------
*  Available Frame Queue Thresholds:
* ------------------------------------------------------------------
* Check that the available frame queue thresholds are not set too
* low.
*
* PARAMETERS:
*       1. 64 bit Minimum LOW threshold (special action commences).
*       2. 64 bit Minimum OK threshold (special action ceases).
*       3. 31 bit Minimum LOW threshold (special action commences).
*       4. 31 bit Minimum OK threshold (special action ceases).
* ------------------------------------------------------------------
*CHECK(Available_Frame_Queue_Thresholds)
*       Severity(High)
*       DATE(20030211)
*       PARMS(400,600,200,400)
*       REASON('System may not recover in time if set too low');
*
* ------------------------------------------------------------------
*  V=R specification:
* ------------------------------------------------------------------
* Check for the existence of V=R (REAL) storage.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Real_Storage_Availability)
*       Severity(Low)
*       DATE(20030102)
*       REASON('Performance may be impacted');
*
```

Figure 3. AOFCUST section HEALTHCHK (Part 6 of 8)

```
* -------------------------------------------------------------------
*  Reconfigurable Storage specification:
* -------------------------------------------------------------------
* Check for the existence of reconfigurable (RSU) storage.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(RSU_Storage_Availability)
*       Severity(Low)
*       DATE(20030102)
*       REASON('Performance may be impacted');
*
* -------------------------------------------------------------------
*   XCF Cleanup value:
* -------------------------------------------------------------------
* Check that the XCF cleanup time is set to a reasonable value to
* hasten the removal of a dead system from the SYSPLEX.
*
* PARAMETERS:
*       1. Recommended XCF cleanup time in seconds.
*          The maximum acceptable value is 86400
* -------------------------------------------------------------------
*CHECK(XCF_Cleanup_Value)
*       Severity(Low)
*       DATE(20030102)
*       PARMS(15)
*       REASON('Quick removal of a dead system from SYSPLEX');
*
* -------------------------------------------------------------------
*  XCF Failure Dectection Interval setting:
* -------------------------------------------------------------------
* Check that the XCF failure detection interval equates to the
* formula PARM1*SPINTIME+PARM2.
*
* PARAMETERS:
*       1. Multiplier.
*       2. Constant.
* -------------------------------------------------------------------
*CHECK(XCF_Failure_Detection_Interval)
*       Severity(Medium)
*       DATE(20030102)
*       PARMS(2,5)
*       REASON('Allow adequate time to recover from spin situation
*before system is assumed dead');
*
* -------------------------------------------------------------------
*  Sysplex Failure Management:
* -------------------------------------------------------------------
* Check that the status of a SYSPLEX failure management (SFM) policy
* is as recommended.
*
* PARAMETERS:
*       1. Recommended SFM status (ACTIVE/INACTIVE).
* -------------------------------------------------------------------
*CHECK(XCF_SYSPLEX_Failure_Management)
*       Severity(Medium)
*       DATE(20030102)
*       PARMS(ACTIVE)
*       REASON('An SFM policy provides better failure management');
*
```

Figure 3. AOFCUST section HEALTHCHK (Part 7 of 8)

```
* ------------------------------------------------------------------
*  SDUMP dynamic allocation of datasets:
* ------------------------------------------------------------------
* Check that SDUMP is using dynamic allocation and that it has
*  not been disabled by the CHNGDUMP command.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(SDUMP_Availability)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('SDUMP setup should ensure adequate diagnostics are
*gathered on the 1st occurrence of problems');
*
* ------------------------------------------------------------------
*  GRS mode:
* ------------------------------------------------------------------
* Check that GRS is in the suggested mode
*
* PARAMETERS:
*       1. Mode required, STAR, RING or NONE.
* ------------------------------------------------------------------
*CHECK(GRS_Mode)
*       Severity(High)
*       DATE(20030102)
*       PARMS(STAR)
*       REASON('GRS should run in STAR mode to improve performance.')
*
* ------------------------------------------------------------------
*  Couple Dataset Separation:
* ------------------------------------------------------------------
* Check that SYSPLEX Couple dataset and Function Couple datasets
*  are properly isolated with alternates.
*
* Parameters:  N/A
* ------------------------------------------------------------------
*CHECK(CDS_Dataset_Separation)
*       Severity(High)
*       DATE(20030102)
*       reason('Ensure that CDS separation has been maintained');
*
* ------------------------------------------------------------------
*  Filesystem Automove setting:
* ------------------------------------------------------------------
* Check that Unix System Services Filesystem Automove is correct
*       in a SYSPLEX environment
* PARAMETERS:
*       1. File system MODE, SYSPLEX has Automove support
*          other File Modes, NOPLEX will not check AutoMove
* ------------------------------------------------------------------
*CHECK(USS_FILESYS_CONFIG)
*       Severity(High)
*       DATE(20030102)
*       PARMS(SYSPLEX)
*       REASON('USS Automove moves a file system to a new system in
*the Sysplex when the owning system fails');
*)
HEALTHCHK(
*CHECK(GRS_Mode)
*       DATE(my date)
*       PARMS(STAR)
*       TIMEINT(24:00)
*       SEVERITY(High)
*       REASON('my reason');
)
```

Figure 3. AOFCUST section HEALTHCHK (Part 8 of 8)

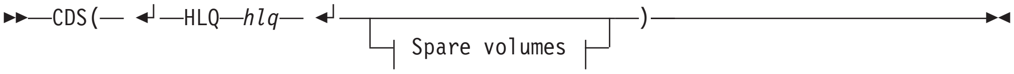
CDS section – spare volumes for CDS recovery

Purpose

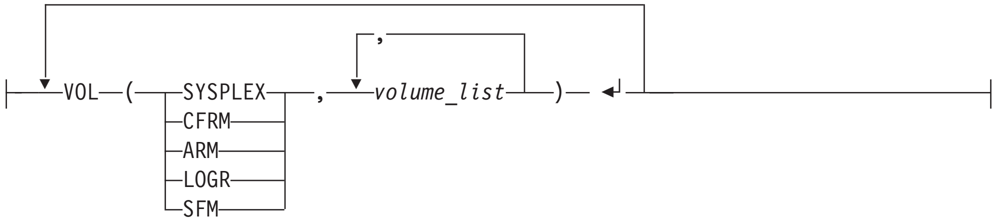
The volumes on which msys for Operations may allocate a new alternate CDS are specified in the CDS section. Every CDS type has its own pool of spare volumes. If you define no spare volumes for a CDS type, no recovery will be performed for this type even if CDS is set in the AUTO section.

The CDS section contains two types of entries, the high level qualifier for the new alternate CDSs, and one list of spare volumes for every CDS type. The list can contain up to eight volumes.

Format



Spare volumes



Parameters

HLQ This value keyword introduces the high-level qualifier for the data sets.

hlq The high-level qualifier can consist of one to three data set qualifiers. The name must comply to the MVS naming rules and can have up to 26 characters.

VOL Every VOL entry specifies the spare volumes for one CDS type. Only one VOL entry can be specified per CDS type.

CDS_type
The CDS type must be the first element of the VOL entry.

volume_list
For every VOL entry, up to eight volumes can be specified. Note that the existence of these volumes is not checked by msys for Operations.

Usage

- The spare volumes must not be managed by SMS.
- In order to improve performance and availability, you should associate spare volumes with CDS types according to the recommendations given in *MVS Setting Up a Sysplex*. It is recommended that you define three spare volumes for every CDS type for which alternate CDSs are to be (re)created automatically.

Example

```
CDS(
  HLQ AOC.CDS.TEST
  VOL (SYSPLEX,AOCLIB,AOCUSR,AOCBCK)
  VOL (CFRM,AOCUSR,AOCLIB,AOCBCK)
  VOL (LOGR,AOCLIB,AOCUSR,AOCBCK)
  VOL (SFM,AOCLIB,AOCUSR,AOCBCK)
  VOL (ARM,AOCLIB,AOCUSR,AOCBCK)
)
```

HW section – hardware configuration

Purpose

The purpose of this function is the cross-validation of the hardware configuration mapped out in AOFCUST against the actual running hardware configuration. This information is critical to accurately control logical partitions (LPAR) on any supported CPC within the HMC/SE LAN over the BCP Internal Interface. Previously, an exposure existed where an operation such as System Reset could be sent to the wrong partition and disrupt operations on that image.

Format

```
►►──HW(── ◄┘──┤ CPC ├──┤ Image ├──)──────────────────────────────────────►◄
```

CPC

```
   ┌──────────────────────────────────────────────────────┐
   ▼                                                      │
├──CPC──(──cpcsyn──,──netid.nau──,──authtkn──)── ◄┘───────┴──────────────────┤
```

Image

```
   ┌────────────────────────────────────────────────────────────────────────┐
   ▼                                                                        │
├──IMAGE──(──sysname──,──lparname──,──cpcsyn──,──plexname──,──┬─CF────┬──)── ◄┘──┴──────┤
                                                              ├─MVS───┤
                                                              └─OTHER─┘
```

Parameters

CPC This keyword defines the Central Processor Complex (CPC), and may be repeated as much as necessary.

cpcsyn This is the synonym character string for the Central Processor Complex (CPC). The maximum length is eight characters.

Note: It is strongly recommended that you use the processor id you defined in HCD as the synonym.

netid.nau
This is the network address of the Support Element. Obtain this information from the CPC Support Element or the HMCs where the CPC is defined.

authtkn
This is the authorization information for communications. The CPC support element, BCP Internal Interface configuration provides this information. The authorization value is the BCP Internal Interface community name. msys for Operations can process *authtkn* specifications in uppercase letters only. Therefore, it may be necessary to change your Support Element BCP Internal Interface community name field, see "Preparing the Support Element" on page 32.

IMAGE
defines a single system and its linkage to a CPC, and may be repeated as much as necessary.

sysname
This is the name of the software image (system). For MVS systems it is the MVS system name that has been defined. For a coupling facility (CF) it is the coupling facility name that has been defined. For other systems use the name that identifies them in your system configuration.

A This name needs to be unique within AOFCUST member being used for an
A msys for Operations instance.

lparname
This is the name of a Logical PARtition (hardware image), the system defined to run on the CPC specified with *cpcsyn*.

Note: When defining a CPC running in basic mode, you must define the LPAR name to be identical to the CPC name.

cpcsyn This is the synonym character string for the Central Processor Complex (CPC). The maximum length is eight characters.

plexname
This is the name of the sysplex where the MVS system *sysname* is defined as a member. If sysname identifies a coupling facility, refer to the coupling definitions to identify the sysplex name that the coupling facility is related to. If *sysname* is not a member of an MVS sysplex, omit the parameter but leave the comma.

MVS|CF|OTHER
Choose MVS if *sysname* has a z/OS or OS/390 operating system. Choose CF if *sysname* is a coupling facility, use OTHER for any other operating system.

Example

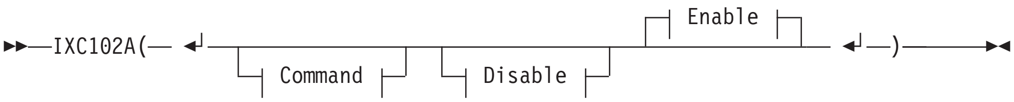
```
HW(
CPC   (FREEWAY,DEIBMD1.X7F1E30A,PUBLIC)
CPC   (SAFOS  ,DEIBMD1.X7F1F20A,PUBLIC)
CPC   (YORAMA ,DEIBMD1.X7E1FA0A,PUBLIC)
IMAGE (CF1     ,CF1     ,FREEWAY ,SYSPLEX ,CF   )
IMAGE (CF2     ,CFF     ,FREEWAY ,SYSPLEX ,CF   )
IMAGE (CFD     ,CFD     ,FREEWAY ,CIM7PLEX,CF   )
IMAGE (CFx     ,CFE     ,FREEWAY ,SYSPLEX ,CF   )
IMAGE (CIM7    ,CIM7    ,FREEWAY ,CIM7PLEX,MVS  )
```

IXC102A section – hardware commands

Purpose

This section allows you to define alternative hardware commands (SYSRESET is the default) to be performed when the automation of messages IXC102A and IXC402D take place. Because the automation flag XCF also controls other functions, you can enable or disable the automation of messages IXC102A and IXC402D for particular systems.

Format



Command

```
   ┌─────────────────────────────────────┐
   ▼                                     │
├──CMD──(──sysname──,──command──)── ◄┘ ──┴───────────────────────────────┤
```

Disable

```
   ┌──────────────────────────────────┐
   │              ┌─,──────┐          │
   ▼              ▼        │          │
├──DISABLE──(────system────┴──)── ◄┘ ─┴──────────────────────────────────┤
```

Enable

```
   ┌─────────────────────────────────┐
   │             ┌─,──────┐          │
   ▼             ▼        │          │
├──ENABLE──(────system────┴──)── ◄┘ ─┴───────────────────────────────────┤
```

Parameters

CMD This keyword defines the hardware command to be sent to the specified system in case this system is being partitioned out of the sysplex by the automation.

sysname
The name of a MVS image.

command
This must be one of the following commands:

1. SYSRESET [CLEAR]—this is the default
2. DEACTIVATE
3. ACTIVATE [P(image_profile_name)]
4. LOAD [P(load_profile_name)] [CLEAR]

   where

   CLEAR
   indicates that the storage is being cleared.

   P specifies the profile to be used. The name can consist of up to 16 alphanumeric characters. If the parameter is omitted the last profile is being used.

> **Note:**
> The following restriction applies to the processor operations commands ACTIVATE and LOAD:
>
> Both commands invoke processor functions, which can cause asynchronous events such as operator messages at BCP (Basic Control Program) internal interface initialization time or processor hardware wait states. Currently, the BCP Internal Interface does not allow to monitor and control these events.

The command is being sent to the Support Element before the outstanding WTOR is replied.

Note: The commands above are disruptive commands. Because the Support Element still considers the affected image as operating, it rejects any disruptive commands unless otherwise stated. This requires that the option FORCE is specified, and it is **automatically** appended by the automation routine.

ENABLE
This keyword defines the system(s) that are enabled for the automation of the IXC102A and IXC402D messages.

DISABLE
This keyword defines the system(s) that are disabled for the automation of the IXC102A and IXC402D messages.

Usage

The following apply to the use of this automation function:

- msys for Operations must be running on all systems in the sysplex because the message is issued only on the system that detects the failure condition.
- The function must be enabled on each system.
- The inoperative system must be running on a G3 or follow-on CPC that has the required MCL installed.
- At least one system, on which the BCP Internal Interface must be available, must have access to the SE of the inoperative system. (See “Hardware prerequisites” on page 30.)

Example

In the following example, the first system is re-IPLed when the automation takes place. The second system is deactivated, and the third system is not automated at all. All systems not specified are automated depending on the flag XCF in the AUTO section. For these systems the default action SYSRESET CLEAR is being performed.

```
IXC102A(
  CMD(sys1,LOAD CLEAR)
  CMD(sys2,DEACTIVATE)
  DISABLE(sys3)
)
```

WTOBUF section – jobs to be canceled in case of buffer shortage

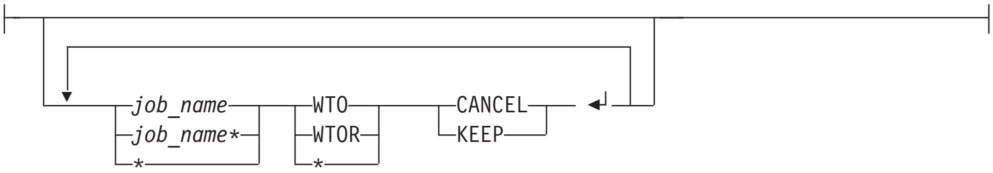
Purpose

In the WTOBUF section, you specify which jobs can be canceled in case of a WTO or WTOR buffer shortage. Every entry of this section states that a job or group of jobs must be kept or canceled when a WTO and/or WTOR buffer shortage occurs.

Format

```
►►──WTOBUF(── ◄┘─┤ Jobs ├──)─────────────────────────────────────────────►◄
```

Jobs



Parameters

job_name

Specifies a job or group of jobs to be kept or canceled. You specify a group of jobs by using an asterisk (’*’) as a placeholder for all jobs or for jobs whose name starts with a given string (for example, ’abc*’ for all job names starting with ’abc’).

buffer_type

Possible values are:

WTO	WTO buffer
WTOR	WTOR buffer
*	Both buffer types

processing_option

Possible values are

CANCEL This keyword indicates that the specified job is canceled when the job caused the shortage and the shortage cannot be relieved within a specific time frame.

KEEP This keyword indicates that the specified job must not be canceled when the job caused the shortage.

Usage

msys for Operations reads the entries of this section downward and applies the first matching entry it finds, regardless of whether more specific matching entries occur farther down in the list. Thus, when the first entry is `* * CANCEL`, and the second is `ABCD * KEEP`, then job ABCD is canceled. All jobs for which no matching entry is found are kept.

Example

In the following example, all jobs whose name starts with ’ABC’ are kept, except job ABCD. All other jobs are canceled.

```
WTOBUF(
  ABCD * CANCEL
  ABC* * KEEP
  *    * CANCEL
)
```

Chapter 4. Making security definitions

This chapter describes how to protect your system resources from unauthorized access and how to implement a security concept that grants operators access to those commands and resources they need. The recommended method to do this in msys for Operations is to use a Security Access Facility (SAF) product, such as Resource Access Control Facility (RACF).

An alternative method to protect your resources and commands without using an SAF product is described in “Defining security using msys for Operations definitions” on page 229.

Activating security classes

An INGSAF1 sample job is delivered with msys for Operations in the SINGSAMP library. The sample predefines security settings for your use which enable you to make a quick start. Customize the job (follow the instructions in the sample) and submit the job. The following assumes that you either tailor the file to your enterprise’s requirements, or that you issue the necessary commands from the system console.

To restrict which operators can log on to the msys for Operations domain, to restrict which commands the operators are authorized to issue, and to ensure that the msys for Operations domain starts properly, the following SAF classes must be activated:

- APPL
- NETCMDS
- NETSPAN

To activate these classes, issue the following commands:

```
SETROPTS CLASSACT(APPL)
SETROPTS CLASSACT(NETCMDS) GRPLIST
SETROPTS CLASSACT(NETSPAN)
```

To protect the msys for Operations domain from unauthorized access, issue the following command to define the msys for Operations domain name to previously defined RACF class APPL:

```
RDEFINE APPL domain_name UACC(NONE)
```

where *domain_name* is the domain name specified in the msys for Operations startup procedure MSOAPROC (INGNVAP0).

Defining operators, passwords, and logon attributes

This section provides the following information about operator security:

- Operator identifiers and passwords
- Operator logon attributes

Defining operator identifiers and passwords

Define a unique operator identifier for each operator who logs on to msys for Operations by changing or adding an ADDUSER statement in sample INGSAF1.

For example, change:

```
ADDUSER OPER1 PASSWORD(user1)
```

to

```
ADDUSER NEWOPER PASSWORD(PW1)
```

In this example, `NEWOPER` is the operator identifier and `PW1` is the initial password. The first time NEWOPER logs on to msys for Operations, the password must be changed.

Note that the names of msys for Operations commands, components, printers (hardcopy logs), terminals, or task identifiers should not be used for operator identifiers. Also, do not use the following reserved keywords:

ALL
DPR
DST
HCL
HCT
LOG
MNT
NNT
OPT
OST
PPT
SYSOP
TCT

An operator can change their password from the msys for Operations logon panel when an SAF product is being used for password authorization.

Defining tasks to RACF

The PPT and SSIR tasks must be defined to RACF.

For the PPT task, change the domain name in INGSAF1 in the following command to match the domain name that is specified in your msys for Operations startup procedure. For example, if the domain name is MSO01 change:

```
ADDUSER domain_namePPT
```

to

```
ADDUSER MSO01PPT
```

For the SSIR task, change the task name you defined (as described in “Defining the name of the automation router task” on page 46) to match the task name in the following command. For example, if the domain name is MSO01 change:

```
ADDUSER domainSIR
```

to

```
ADDUSER MSO01SIR
```

Use the ADDUSER and ALTUSER commands for all of the msys for Operations autotasks exactly as specified in sample INGSAF1.

Defining operator logon attributes in the NVSS segment of an SAF product

Operator logon attributes describe logon attributes that are associated with an operator. Operator logon attributes are defined in the NVSS segment of an SAF product. The following operator attributes are supported:

CTL
CTL defines an operator’s authority to control resources. CTL(GLOBAL) is recommended to enable operators to control all msys for Operations resources.

IC
IC specifies the initial command list to run when an operator logs on. Initial command list LOGPROF1 defines PF key definitions and a unique MVS console name. It is recommended to use LOGPROF1 for all operators except automated operators (autotasks).

MSGRECVR
MSGRECVR specifies whether an operator is eligible to receive unsolicited messages that are not routed to a particular operator using either the ASSIGN command or msys for Operations automation. Possible values for MSGRECVR are either `YES` or `NO`. Usually, the first operator to be logged on with MSGRECVR(YES) is the authorized receiver of unsolicited messages.

In RACF, operator logon attributes are defined using the ALTUSER command. For example:

```
ALTUSER NEWUSER NETVIEW(IC(LOGPROF1) CTL(GLOBAL) MSGRECVR(YES))
```

When the operator NEWUSER logs on:
- The initial command list LOGPROF1 is run
- Operator NEWUSER controls all msys for Operations resources
- Operator NEWUSER can receive unsolicited messages

Use the RACF PERMIT command to enable operators to log on to the domain name specified in your msys for Operations startup procedure. For example, to enable operator NEWOPER to log on to domain MSO01, specify the following RACF command in INGSAF1:

```
PERMIT MSO01 CLASS(APPL) ID(NEWOPER) ACCESS(READ)
```

where MSO01 is the domain name and NEWOPER is the operator ID. Instead of an operator ID you can also specify a group, for example MSYSOPS1.

Command authorization

This section provides the following information about command authorization:
- Overview
- Command authorization using an SAF product such as RACF
- Command authorization for specific commands
- Additional recommendations about command authorization

Overview

Command authorization is the process of protecting commands from unauthorized use and selectively grant access to them.

Use RACF, or a comparable SAF product, to restrict access to commands, keywords, and values, and to grant operator access to them. You do this by defining the commands, keywords, and values as resources in the NETCMDS class of the SAF product, and then selectively granting operator access. This is most easily done by grouping operators into groups that correspond to their roles and level of expertise. The next step is to connect your users to those groups.

You can also specify commands, keywords, and values that are accessible universally. When you make these changes, you can have them take effect by requesting the SAF product to refresh the NETCMDS class definitions. You do not have to issue msys for Operations commands to include the changes.

In the INGSAF1 sample the following groups are predefined:

MSYSOPS0
Users that are listed in this group are allowed to execute administrative commands.

MSYSOPS1
Users that are listed in this group are allowed to execute FORCE and REBUILD actions on structures.

MSYSOPS2
Users that are listed in this group are allowed to execute the SETXCF command with parameter ACOUPLE and PSWITCH

MSYSOPS3
Users that are listed in this group are allowed to execute the full functionality of INGCF ENABLE and INGCF DRAIN

MSYSOPS4
Users that are listed in this group are allowed to execute most restricted base NVSS commands

A **MSYSOPS5**
A Users that are listed in this group are allowed to execute the CLOSE and
A ACF COLD commands.

Command authorization using an SAF product

The following sections describe the steps that are necessary for authorizing commands using an SAF product such as RACF.

Defining msys for Operations commands as NETCMDS resources

To define msys for Operations commands as resources in the NETCMDS commands class, use resource names as described in the following.

Commands are checked separately from keywords and values. When defining resource names, remember that the command is checked first. Commands, keywords, and value combinations are checked in the following order:

```
netid.luname.command
netid.luname.command.keyword
netid.luname.command.keyword.value
```

Where:

netid
Indicates the VTAM network identifier. You can specify a generic character (*) for this field.

This value is compared with the VTAM network identifier from the last activation of VTAM or +NONE+ if VTAM has not been activated. If you do not need to differentiate between *netid*s and are not concerned about whether VTAM has been active, specify a generic character (*) for this field.

luname
Indicates the domain name for an instance of an msys for Operations program.

command
Indicates the command name on the CMDMDL statement in the DSICMSYS member of DSIPARM, or a command list name. This must be the actual command name and not a synonym defined by the CMDSYN statement. No checking is done to validate that *command* is a valid command or command list name.

keyword
Indicates the keyword identifier which is protected.

value
Indicates the value identifier which is protected when used with the keyword on the command.

The keyword or value used with the command may not match the keyword or value being protected because of synonyms, defaults, and substitutions of values in the resource name.

Examples of NETCMDS resource definitions: The following examples show how to define NETCMDs resources using the RDEFINE command of RACF. RDEFINE can be abbreviated to RDEF.

- Create one resource in the NETCMDS class for each command you want to protect. To protect the CF keyword of the INGCF command, specify the following in the INGSAF1 sample:

  ```
  RDEF NETCMDS *.*.INGRCCHK.INGCF.CF UACC(NONE)
  ```

  INGRCCHK is an internal routine that is called by the INGCF and INGPLEX commands and needs to be specified.
- To minimize the number of occasions where SAF cannot achieve a command authorization decision, you can universally grant or deny access to the remaining commands, keywords, and values by defining a generic resource name for msys for Operations. Using RACF, you can do this for a *netid* of NETA and an *luname* of MSO01 by issuing one of the following commands:

  ```
  RDEFINE NETCMDS *.*.* UACC(READ)
  RDEFINE NETCMDS *.*.* UACC(NONE)
  ```

  To allow an msys for Operations operator to issue an msys for Operations command protected in the NETCMDS class, you must grant a level of access of at least READ.

Summary–how to perform command authorization using SAF

The following step-by-step procedure shows an example (based on the INGSAF1 sample file) of how to define operator authority to RACF, assuming the operators are already defined to RACF.

1. To activate the NETCMDS class, if not already active, specify the following:

   ```
   SETROPTS CLASSACT(NETCMDS) GRPLIST
   ```

2. To define the NETCMDS class as a GENERIC class to allow the use of generic characters, if generic characters will be used, specify the following:

   ```
   SETROPTS GENERIC(NETCMDS)
   ```

3. To define groups of operators, specify the following:

   ```
   ADDGROUP MSYSOPS1
   CONNECT user1 GROUP(MSYSOPS1) UACC(READ)
   CONNECT user2 GROUP(MSYSOPS1) UACC(READ)
   ADDGROUP MSYSOPS2
   CONNECT user3  GROUP(MSYSOPS2) UACC(READ)
   CONNECT user4  GROUP(MSYSOPS2) UACC(READ)
   ADDGROUP MSYSOPS3
   ```

```
CONNECT user5  GROUP(MSYSOPS3) UACC(READ)
CONNECT user6  GROUP(MSYSOPS3) UACC(READ)
ADDGROUP MSYSOPS4
CONNECT user7  GROUP(MSYSOPS4) UACC(READ)
CONNECT user8  GROUP(MSYSOPS4) UACC(READ)
ADDGROUP MSYSOPS5
CONNECT user9  GROUP(MSYSOPS5) UACC(READ)
CONNECT user10 GROUP(MSYSOPS5) UACC(READ)
```

4. Define the commands, keywords and values to be protected.

   To define the software-related functions of INGPLEX and INGCF as resources in the NETCMDS class, specify the following:

   ```
   RDEFINE NETCMDS *.*.INGRRCHK.INGCF.STR UACC(NONE)
   RDEFINE NETCMDS *.*.INGRRCHK.INGPLEX.CDS UACC(NONE)
   RDEFINE NETCMDS *.*.INGRRCHK.INGCF.CF UACC(NONE)
   RDEFINE NETCMDS *.*.INGRRCHK.INGPLEX.HW UACC(NONE)
   ```

5. To associate operator groups with command resources, specify the following:

   ```
   PERMIT *.*.INGRRCHK.INGCF.STR CLASS(NETCMDS) ID(MSYSOPS1) ACC(READ)
   PERMIT *.*.INGRRCHK.INGPLEX.CDS CLASS(NETCMDS) ID(MSYSOPS2) ACC(READ)
   PERMIT *.*.INGRRCHK.INGPLEX.CDS CLASS(NETCMDS) ID(MSYSOPS3) ACC(READ)
   PERMIT *.*.INGRRCHK.INGCF.CF CLASS(NETCMDS) ID(MSYSOPS3) ACC(READ)
   PERMIT *.*.INGRRCHK.INGPLEX.HW CLASS(NETCMDS) ID(MSYSOPS4) ACC(READ)
   ```

   In this example:

   - User1 and user2 can only rebuild or force a selected structure with INGCF STRUCTURE
   - User3 and user4 are allowed to rebuild or force a selected structure, to switch policies and to manipulate couple data sets
   - User5 and user6 can rebuild and force a selected structure, set the sender path of a coupling facility ONLINE or OFFLINE, switch policies, manipulate couple data sets.

Protecting immediate commands when CMDAUTH=SAF

Immediate commands are host msys for Operations commands which are defined with TYPE=I on the CMDMDL statement in DSICMSYS, or defined with TYPE=B on the CMDMDL statement in DSICMSYS and have been entered from the command line. These commands are run under the control of an IRB exit. This environment prohibits the use of an SAF RACROUTE macro to call a security product.

For msys for Operations, the only immediate command that must be protected is the CLOSE IMMED command. To define which operator can stop msys for Operations by issuing the CLOSE IMMED command, add the operator identifier to the sample backup table, CNMSBAK1, provided with msys for Operations in the CNMSAMP library. Copy CNMSBAK1 to your DSIPARM library. For example, to protect the CLOSE command keep the statements predefined in sample member CNMSBAK1:

```
GROUP MSYSOPS5 NETOP1, NETOP2
PROTECT   *.*.CLOSE
```

To authorize the MSYSOPS5 group to use the CLOSE command keep the statement in the CNMSBAK1 sample:

```
PERMIT MSYSOPS5 *.*.CLOSE
```

Additional recommendations for command authorization

Recommended commands to protect

Customers must decide which commands to protect based on their specific security requirements. However, it is recommended that you restrict the following commands, because they can affect the msys for Operations environment or access to it:

- AFTER (Use of the PPT keyword.)
- AT (Use of the PPT keyword.)
- AUTOTBL
- CHRON (Use of the ROUTE keyword. See "Defining security for the CHRON command" on page 84 for more information.)
- CLOSE
- DEFAULTS
- EVERY (Use of the PPT keyword)
- EXCMD. (See "Defining EXCMD command authorization" on page 88 for more information.)
- FOCALPT
- GETCONID
- GLOBALV
- INGRCCHK
- MODIFY
- MVS (See "Defining additional MVS command authority" on page 88 for more information.)
- OVERRIDE
- PURGE
- READSEC (See "NVSS READSEC and WRITESEC commands" on page 92 for more information.)
- REFRESH
- RMTCMD
- RUNCMD
- SUBMIT (See "Defining SUBMIT command authorization" on page 89 for more information.)
- SETCONID
- START
- STOP
- VARY
- WRITESEC (See "NVSS READSEC and WRITESEC commands" on page 92 for more information.)

Exceptions to command authorization checking

Major exceptions to command authorization checking include:

- Commands entered as replies to the msys for Operations WTOR (message DSI802A) are not authority checked. To prevent users from issuing commands using the WTOR, specify CMDWTOR=NO in the MVSPARM statement in DSIDMN. This prevents msys for Operations from issuing the WTOR.
- Command authority checks are not made against the PPT or DST tasks. Therefore, you need not authorize these tasks to access your protected commands.
- Commands issued from a source ID of `*BYPASS*` are not checked for command authorization by:
  - The SAF product OPERCMDS class
  - The SAF product NETCMDS class

The SOURCEID will default to `*BYPASS*` if the command was entered at an extended multiple console support (EMCS) console and the operator was not logged on to the EMCS console.

Auditing command authority checking

You can audit access to protected commands, keywords, and values. This auditing can be done on an individual command, keyword, or value basis.

You can audit access to SAF-defined resources. You can control this auditing on a resource basis. For each resource, you can specify whether to perform no auditing, to audit authorization failures, to audit authorization successes, or to audit all access attempts whether successful or not. For RACF, the auditing level is specified using the RDEFINE or RALTER commands when you define the resource name. Additionally, to allow msys for Operations commands in general to be audited, you must ensure that the RACF SETROPTS statement specifies AUDIT(NETCMDS). RACF generates SMF records that contain details at the audit level you specify for commands. You can then use the RACF report writer to create reports that describe attempts to access RACF-protected resources. For more information on the RACF report writer, refer to the RACF library.

The more auditing you request the SAF product to perform, the more system resources are required by the SAF product. You need to determine the value of the audit level you choose versus the expense in system overhead, both processor and DASD.

Protecting commands containing special characters

There are some special characters that cannot be included in the command identifier or SAF resource name. For this reason, msys for Operations translates these special characters to other characters before passing them to either the msys for Operations command authorization table or the SAF product. The special characters that are translated along with their translated results are:

```
Reserved Character    Translated Result
   .                      /
   *                      +
   %                      ?
   &                      :
   -  (dash)              _  (underscore)
  ' ' (blank)             _  (underscore)
```

For example, the following msys for Operations command can be entered by an msys for Operations operator:

```
LIST MEMSTAT=.*
```

To restrict access to this keyword and value using RACF, include the following RACF profile:

```
LIST MEMSTAT=.+
```

Note that the asterisk was translated to a plus.

Determining the user identity used for authority checking commands

Authority checking restricts the ability of an operator or a task to use commands, keywords, and values.

Determining the user identity used for authority checking commands

Table 4 identifies the operator or task identifier that is used to authority check msys for Operations commands based on the command and environment. The identity is referred to as the SOURCEID.

Table 4. SOURCEID determination

Command and environment	SOURCEID determination
EXCMD command or a same-domain LABEL command prefix used to queue an imbedded command to another task.	The SOURCEID is the task that issued the EXCMD command, or the existing SOURCEID at the time the EXCMD command was issued.
TIMER commands that are scheduled to run under the PPT.	The SOURCEID is the task that issued the AT, EVERY, CHRON, or AFTER command, or the existing SOURCEID at the time the AT, EVERY, CHRON, or AFTER command was issued. **Note:** The SOURCEID is not destroyed by saving and restoring timer commands.
msys for Operations SUBMIT command for jobs submitted to the operating system from msys for Operations.	If OPERSEC=SAFDEF or OPERSEC=SAFCHECK, the identity that is checked by the operating system is the issuer of the SUBMIT command, or the existing SOURCEID at the time the SUBMIT command was issued. For other values of OPERSEC, msys for Operations' authority is used for submitting the job.
msys for Operations commands that were entered at an MVS operator console.	When an MVS console has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, msys for Operations commands can be entered from that MVS console. This is done by prefixing the msys for Operations command with the msys for Operations designator character, which by default is %. If the MVS operator has logged on to the MVS console with a user ID, the SOURCEID is the user ID of the MVS operator. If an operator has not logged on at the EMCS console, the SOURCEID of that task defaults to `*BYPASS*`. Commands issued from a source ID of `*BYPASS*` are not checked for command authorization by: • The msys for Operations command authorization table • The SAF product OPERCMDS class • The SAF product NETCMDS class **Note:** If a command is entered from the MVS master console, it will be routed to one of the following: • The autotask with the specific console name • The autotask with console name "*MASTER*" • The autotask with console name "*ANY*"

Table 4. SOURCEID determination (continued)

Command and environment	SOURCEID determination
msys for Operations commands that are entered using the MVS MODIFY command.	When an MVS console has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, msys for Operations commands can be entered from that MVS console by issuing an MVS MODIFY or STOP command against the msys for Operations task. The msys for Operations command is entered as text following the MODIFY command. The first parameter on the MODIFY command is the application ID that is being modified. If the MVS operator has logged on to the MVS console with a user ID, the SOURCEID is the user ID of the MVS operator. If an operator has not logged on at the EMCS console, the SOURCEID of that task defaults to *BYPASS*. Commands issued from a source ID of *BYPASS* are not checked for command authorization by: • The msys for Operations command authorization table • The SAF product OPERCMDS class • The SAF product NETCMDS class **Note:** If a command is entered from the MVS master console, it will be routed to: • The autotask with the specific console name • The autotask with console name "*MASTER*" • The autotask with console name "*ANY*"
msys for Operations commands that were entered by TSO users.	When a TSO user ID has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, msys for Operations commands can be entered from that TSO user ID when the user is acting as an MVS operator by using an EMCS console session, or when using SDSF. The SOURCEID is the TSO user's user ID.
Commands issued from JCL.	When a job that issues a msys for Operations command is submitted by a TSO user ID, the SOURCEID is the TSO user ID. If the ID of the submitter is unknown, a default user ID is inserted. The value of the default user ID is defined by the system installation.
MVS ROUTE command issued from msys for Operations.	If the MVS command ROUTE is issued from a msys for Operations task, the originating source ID is always passed to the SAF product for authorization checks in the OPERCMDS class. This occurs for all settings of AUTHCHK and CMDAUTH.
Commands that are routed to an operator from the automation table.	The SOURCEID is the operator ID to which the command is routed. **Note:** Commands from the automation table are subject to authority checking unless SEC=BY was specified on the CMDMDL statement or SEC=DE was specified (or SEC was not specified) and AUTOSEC=BYPASS is in effect. For more information, refer to the DEFAULTS command in the msys for Operations online help.

Understanding security for specific commands

This section provides additional information about protecting the following commands:

- CHRON
- EXCMD
- MVS
- SUBMIT

Defining security for the CHRON command

The CHRON command has a syntax that is more complex than most commands. CHRON uses multiple levels of keywords, items in lists, and quoted strings.

Command security for the CHRON command is checked so that operands within parentheses can be uniquely defined in an SAF product.

The following rules describe CHRON commands and which command identifiers are checked:

RULE 1: Each keyword that does not take a value (NOSAVE, SAVE, LOCAL, GMT, REFRESH, TEST, and DEBUG) is checked in the form:

Command example:

```
netid.luname.CHRON.keyword
```

RULE 2: Each keyword with a value is checked in the form:

```
netid.luname.CHRON.keyword.value
```

With the CHRON command, the value may be a list or quoted string.

Command example:

```
CHRON AT=(),RECOVERY=IGNORE,NOSAVE,LOCAL,ROUTE=OPER1,ID=TEST1,
COMMAND='MSG ALL HELLO'
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.AT.()
netid.luname.CHRON.RECOVERY.IGNORE
netid.luname.CHRON.NOSAVE
netid.luname.CHRON.LOCAL
netid.luname.CHRON.ROUTE.OPER1
netid.luname.CHRON.ID.TEST1
netid.luname.CHRON.COMMAND.'MSG_ALL_HELLO'
```

Rule 3A: Keywords appearing within parenthesized lists of other keywords are checked using the hierarchy of keywords with a ″(″ between so that the keyword hierarchy can be uniquely identified. The compound keyword that is generated is tested with the value of the innermost keyword. This checking is done at each level of the nesting of the lists. When a keyword is within a list that is the value of another keyword, the notation uses both keywords with a ″(″ between them.

Rule 3B: From the outermost to innermost, if a ″keyword=(list)″ appears, if any values appear in the list without keywords, the ″keyword=value″ check is done for that value. The keyword that is checked is the keyword hierarchy defined by Rule 3A.

Command example:

```
CHRON EVERY=(INTERVAL=(000-01.00.00 FOR=08.00.00))
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.EVERY.(INTERVAL=(000_01/00/00_FOR=08/00/00))
netid.luname.CHRON.EVERY(INTERVAL.(000_01/00/00_FOR=08/00/00)
netid.luname.CHRON.EVERY(INTERVAL.000_01/00/00
netid.luname.CHRON.EVERY(INTERVAL(FOR.08/00/00
```

Substitution of certain special characters is performed as described in “Protecting commands containing special characters” on page 82. For example, a dash becomes an underscore in the command identifier.

Rule 4: Quoted string values are checked as a single value, including the apostrophes and all text within the apostrophes.

Command example:

```
netid.luname.CHRON.REM.'ISN''T THIS A REMARK STRING?'
```

The following command identifier is checked:

```
CHRON REM='ISN''T THIS A REMARK STRING?'
```

Rule 5: For the DAYSWEEK keyword, days of the week can be followed by a sublist identifying particular weeks of the month. The day name and each item in the sublist are treated as a unit.

Command example:

```
CHRON EVERY=(DAYSWEEK=(NOT MON(1ST 2nd)))
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.EVERY.(DAYSWEEK=(NOT_MON(1ST_2ND)))
netid.luname.CHRON.EVERY(DAYSWEEK.(NOT_MON(1ST_2ND))
netid.luname.CHRON.EVERY(DAYSWEEK.NOT
netid.luname.EVERY(DAYSWEEK.MON(1ST)
netid.luname.EVERY(DAYSWEEK.MON(2ND)
```

This lets you check the sublist values without concern for the order of the items within the sublist. Notice that the value ″MON(1st 2nd)″ is not checked since the values MON(1st) and MON(2nd) are checked.

The following table illustrates a detailed list of possible command identifiers that may be defined for the CHRON command. The rule that causes the command identifier to be checked is shown in the second column.

Table 5. Command identifiers for the CHRON command

Commands and keywords identifier	RULE	SAF resource identifier
CHRON	Command Name	netid.luname.CHRON
AT=		
	2	netid.luname.CHRON.AT.()
	2	netid.luname.CHRON.AT.(timespec datespec) [2]
	3B	netid.luname.CHRON.AT.timespec
	3B	netid.luname.CHRON.AT.datespec[2]
	2	netid.luname.CHRON.AT.yyy_mm_dd_hh/mm/ss/micros[2]
AFTER=	2	netid.luname.CHRON.AFTER.timespec [2]
	2	netid.luname.CHRON.AFTER.ddd_hh/mm/ss/micros [2]
EVERY=	2	netid.luname.CHRON.EVERY.NONE
	2	netid.luname.CHRON.EVERY.( )
	2	netid.luname.CHRON.EVERY.(everyoptions) [2]

Table 5. Command identifiers for the CHRON command (continued)

Commands and keywords identifier	RULE	SAF resource identifier
EVERY=(INTERVAL=	3A 3B 3B 3A	netid.luname.CHRON.EVERY(INTERVAL.( ) netid.luname.CHRON.EVERY(INTERVAL.(intervaloptions) [2] netid.luname.CHRON.EVERY(INTERVAL.timespec [2] netid.luname.CHRON.EVERY(INTERVAL. ddd_hh/mm/ss/micros [2]
EVERY=(INTERVAL= (FOR=	3A 3A	netid.luname.CHRON.EVERY(INTERVAL(FOR.timespec netid.luname.CHRON.EVERY(INTERVAL(FOR. hh/mm/ss/micros [2]
EVERY=(INTERVAL= (MXREPEAT=	3A 3A	netid.luname.CHRON.EVERY(INTERVAL(MXREPEAT. NOLIMIT netid.luname.CHRON.EVERY(INTERVAL(MXREPEAT. repeat_count
EVERY=(INTERVAL= (OFF=	3A 3A	netid.luname.CHRON.EVERY(INTERVAL(OFF.timespec netid.luname.CHRON.EVERY(INTERVAL(OFF. hh/mm/ss/micros [2]
EVERY=(REMOVE=	3A 3A, 3B 3B 3B 3A	netid.luname.CHRON.EVERY(REMOVE.MANUALLY netid.luname.CHRON.EVERY(REMOVE.(removeoptions) [2] netid.luname.CHRON.EVERY(REMOVE.datespec [2] netid.luname.CHRON.EVERY(REMOVE.timespec [2] netid.luname.CHRON.EVERY(REMOVE. yyyy_mm_dd_hh/mm/ss/micros [2]
EVERY= (REMAFTER=	3A 3A	netid.luname.CHRON.EVERY(REMAFTER.timespec [2] netid.luname.CHRON.EVERY(REMAFTER. ddd_hh/mm/ss/micros [2]
EVERY= (DAYSWEEK=	3A 3B 3B 3B 5	netid.luname.CHRON.EVERY(DAYSWEEK.ALL netid.luname.CHRON.EVERY(DAYSWEEK.(daysweeklist) [2] netid.luname.CHRON.EVERY(DAYSWEEK.NOT netid.luname.CHRON.EVERY(DAYSWEEK.dayname netid.luname.CHRON.EVERY(DAYSWEEK. dayname (sublist_element) [2]
EVERY=(DAYSMON=	3A 3B 3B 3B	netid.luname.CHRON.EVERY(DAYSMON.ALL netid.luname.CHRON.EVERY(DAYSMON.(dayslist) [2] netid.luname.CHRON.EVERY(DAYSMON.NOT netid.luname.CHRON.EVERY(DAYSMON.dayofmonth [2]
EVERY=(CALENDAR=	3A 3B 3B 3B	netid.luname.CHRON.EVERY(CALENDAR.ALL netid.luname.CHRON.EVERY(CALENDAR.(calendarlist) [2] netid.luname.CHRON.EVERY(CALENDAR.NOT netid.luname.CHRON.EVERY(CALENDAR.keyname [2]
RECOVERY=	2 2 2	netid.luname.CHRON.RECOVERY.IGNORE netid.luname.CHRON.RECOVERY.AUTOLGN netid.luname.CHRON.RECOVERY.PURGE
SAVE	1	netid.luname.CHRON.SAVE

Table 5. Command identifiers for the CHRON command (continued)

Commands and keywords identifier	RULE	SAF resource identifier
NOSAVE	1	netid.luname.CHRON.NOSAVE
LOCAL	1	netid.luname.CHRON.LOCAL
ID=	2	netid.luname.CHRON.ID.idname
NOTIFY=	2	netid.luname.CHRON.NOTIFY.(notifylists)
NOTIFY=(PURGE=	3B 3B	netid.luname.CHRON.NOTIFY(PURGE.(purgelist) netid.luname.CHRON.NOTIFY(PURGE.taskname
NOTIFY=(REMOVE=	3B 3B	netid.luname.CHRON.NOTIFY(REMOVE.(removelist) netid.luname.CHRON.NOTIFY(REMOVE.taskname
NOTIFY=(IGNORE=	3B 3B	netid.luname.CHRON.NOTIFY(IGNORE.(ignorelist) netid.luname.CHRON.NOTIFY(IGNORE.taskname
NOTIFY=(RUN=	3B 3B	netid.luname.CHRON.NOTIFY(RUN.(runlist) netid.luname.CHRON.NOTIFY(RUN.taskname
REFRESH	1	netid.luname.CHRON.REFRESH
TEST	1	netid.luname.CHRON.TEST
DEBUG	1	netid.luname.CHRON.DEBUG
COMMAND=	4	netid.luname.CHRON.COMMAND.’quoted string’ [2]
REM=	4	netid.luname.REM.’quoted string’ [2]

Defining EXCMD command authorization

The msys for Operations EXCMD command is used to send commands to another task.

There are two operands that are used when issuing the EXCMD command. One is the *operator_id* where the command is being sent, and the other is the *command* being sent. These two operands are checked as a keyword-value pair.

Note: When protecting the target verb of EXCMD, specify the command verb, not any synonym. Unless otherwise documented, the verb is the label used on the CMDMDL statement. The verb for labeled commands beginning with a slash is EXCMD.

For example, the command identifier to protect `EXCMD OPER1 LOGOFF` is:

```
PROTECT *.*.EXCMD.OPER1.LOGOFF
```

For information about the EXCMD command refer to the online help.

Defining additional MVS command authority

You can protect individual MVS system commands from unauthorized use with the OPERCMDS class of an SAF product, such as RACF. This is additional

[2] This value may have a special character, such as ″.″ or ″-″, for example in the programmer time notation. You substitute the character ″/″ for ″.″ and ″_″ for ″-″ when making the security definition.

authorization checking done at the MVS level, after the command security checking done by the NETCMDS class of an SAF product.

To protect MVS commands:

1. Ensure your OPERSEC setting has a value of SAFCHECK or SAFDEF.
2. Define command profiles to restrict specific commands from operators. For example, to restrict all operators from being able to issue an MVS QUIESCE command, enter:
   ```
   RDEFINE OPERCMDS MVS.QUIESCE UACC(NONE)
   ```
3. Ensure that the OPERCMDS class is active and enabled for processing. The following RACF commands can be used to do this:
   ```
   SETROPTS CLASSACT(OPERCMDS)
   SETROPTS RACLIST(OPERCMDS)
   ```
4. When the OPERCMDS class is active, use the RACF REFRESH function when you change a definition:
   ```
   SETROPTS RACLIST(OPERCMDS) REFRESH
   ```

Defining SUBMIT command authorization

You can protect jobs submitted from NVSS using the SUBMIT command. When the NVSS SUBMIT command is issued, you have three layers of protection that you can use:

1. The SUBMIT command can be protected using NVSS command authorization. This is your first layer of protection. By protecting at this level, you can stop the processing for unauthorized users before the job is ever submitted to the system.
2. For jobs that reside in data sets that are NOT part of the DSIPARM data set concatenation, you can use the SAF DATASET class to prevent users from accessing those data sets. Using the SAF DATASET class prevents users from submitting jobs that are members of those data sets. This is the second layer of protection. An attempt to access the data set is made before the job is actually submitted.
3. The SAF JESJOBS class can be used to prevent users from submitting specific jobs. This is effective for DSIPARM and non-DSIPARM data sets. This is the third layer of protection. This check happens after the job has been submitted to JES, (not synchronously with the msys for Operations SUBMIT command).

   Note: A failure at this level will not be reported back to the msys for Operations console. The JESJOBS class failure is only reported to the master console or the syslog.

Controlling access to data sets and members

To prevent unauthorized changes of data, you can protect data sets with an SAF product, such as RACF. To prevent unauthorized viewing of passwords and other restricted information, protect them with msys for Operations commands such as READSEC and WRITESEC. See “NVSS READSEC and WRITESEC commands” on page 92 for recommendations.

Data set security

You can restrict unauthorized alteration of data sets from the msys for Operations environment using the DATASET class of the security product. The following are some considerations when using the DATASET class of the security product:

- msys for Operations requires CONTROL access to the DSILOG data set to write to the netlog.
- msys for Operations requires READ access to the first data set identified by the DSILIST DD statement.
- msys for Operations requires READ access to non-DSIPARM data sets that are specified with the msys for Operations SUBMIT command.
- Each of the following NVSS commands require UPDATE access to the first data set identified by the DSILIST DD statement.
  - AUTOTBL (with the LISTING keyword)
  - AUTOCNT (with the FILE keyword)
  - QRYGLOBL (with the FILE keyword)

Note: NVSS trace records are not made for calls to the DATASET class, because the calls are made by MVS for the NVSS tasks.

Restricting and granting access to data sets

To activate the data set protection described in the preceding section, do the following:

1. Add profiles for the data sets you want to protect. The RACF product requires that the highest-level qualifier of the data set name be either a task or group name.

   For example, use the RACF ADDSD command to add data set profiles. From an authorized TSO user, enter the following command to protect the OPER1.STATS data set:

   ```
   ADDSD 'OPER1.STATS'
   ```
2. Authorize the operator tasks so they can access the data set. For example, use the RACF PERMIT command to authorize operator tasks to the data set. To authorize NETOP1 to have update access to OPER1.STATS, enter the following command from an authorized TSO user ID:

   ```
   PERMIT 'OPER1.STATS' CLASS(DATASET) ID(NETOP1) ACCESS(UPDATE)
   ```

Granting NVSS and the STC-user access to data sets

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 NVSS 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 NVSS address space. This requires the RACF ALTER access to these data sets for NVSS (*nvuserid*).

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. To add a data set profile specify the following:

```
ADDSD 'hlq.*.HSA*.*' UACC(NONE)
```

where:

hlq
is the high-level qualifier of the automation status file used by the current NVSS

HSA*
is the NVSS domain

To grant the *stcuser* and the *nvuserid* access to the data set, specify, for example:

```
PE 'hlq.*.**.*' CLASS(DATASET) ACC(UPDATE) ID(stcuser)
PE 'hlq.*.**.*' CLASS(DATASET) ACC(ALTER) ID(nvuserid)
```

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 NVSS address space. This requires the RACF ALTER access to these data sets for NVSS.

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 NVSS

X O or P

For an example of how to do your RACF definitions refer to “Access to XCF utilities” on page 90.

Access to IPL Information: The new automation function collecting, displaying, comparing, and deleting IPL information uses two started tasks. The first started task runs immediately after an IPL as part of COMMNDxx list processing, and collects the IPL information in the msys for Operations VSAM data set "IPLDATA". The remaining functions are handled by a NVSS 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 NVSS 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 NVSS address space. This requires the RACF ALTER access to these data sets for NVSS.

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 NVSS

opid is the NVSS operator ID

x L, N, or O

For an example of how to do your RACF definitions refer to “Access to XCF utilities” on page 90.

Access to CDS spare volumes: 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 data set names are created as follows:

```
hlq123.cdstype.CDSnn
```

where:

hlq123 is the high-level qualifier defined during customization

cdstype is ARM, CFRM, LOGR, SFM, SYSPLEX

To create a data set profile and to grant the *stcuser* access specify, for example:

```
ADDSD 'hlq.**.**' UACC(NONE)
PE 'hlq.**.**' CLASS(DATASET) ACC(ALTER) ID(stcuser)
```

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. To add a user-defined CDS profile and grant the *stcuser* access specify, for example:

```
ADDSD 'sys1.cds.**' UACC(NONE)
PE 'sys1.cds.**' CLASS(DATASET) ACC(READ) ID(stcuser)
```

When LOGGER recovery is enabled, the user ID needs RACF ALTER access to the LOGR couple data sets as well. For example, specify the following:

```
ADDSD 'sys1.cds.log.**' UACC(NONE)
PE 'sys1.cds.log.**' CLASS(DATASET) ACC(ALTER) ID(stcuser)
```

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

For an example of how to do your RACF definitions refer to “Access to XCF utilities” on page 90.

NVSS READSEC and WRITESEC commands

Use the msys for Operations READSEC and WRITESEC commands to restrict access to data sets and members by msys for Operations commands. When you specify security for the READSEC command, it affects all of the msys for Operations commands which can display sensitive information, such as:

- BROWSE with a member name
- NCCF LIST with the CLIST or PROFILE keywords

- PIPE stages
  - < (From disk)
  - QSAM
- VSAM command DSIVSMX

Using READSEC and WRITESEC is the only way to prevent operators from viewing data sets and members using these msys for Operations commands. In msys for Operations, security is defined so that operators have access to DSIOPEN and msys for Operations online help. DSIOPEN is a DD name designed to hold information which should not be secured, such as NEWS data and PF key definitions. Anything other than DSIOPEN and online help may be considered sensitive information.

Because attempts to define security for these commands is considered a severe error, message BNH115A is generated every time an operator logs on. The error text for this message is “SPECIAL SECURITY IN EFFECT FOR BROWSE AND READSEC”, which indicates msys for Operations has defined default protection for sensitive data sets and members, and the msys for Operations commands which display data sets or members will fail. You must delete any security definitions for the commands and reinitialize msys for Operations to clear the error condition.

If you use command authorization without specifying values for READSEC and WRITESEC, operators will have access to all data sets and members.

Do not protect DD name CNMPNL1, operators need to access online help that is contained there.

For more information about how to use the READSEC and WRITESEC commands, refer to the online help.

Controlling access to the processor hardware functions

The following describes what must be defined in a SAF product such as RACF to enable the usage of the SNMP over the BCP Internal Interface, in the following referred as HSAET32. Note that this interface is needed if you have enabled CF and XCF automation in your AOFCUST definitions.

Enabling NVSS

Before using msys for Operations for CF or XCF automation, NVSS must be authorized for HSAET32.

1. Define resource HSA.ET32OAN.HSAET32 in the CLASS FACILITY
2. Permit NVSS READ ACCESS to this facility class resource. The following example shows the RACF commands used to define the resource and to grant READ access to it for NVSS.

```
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY)
RDEFINE FACILITY HSA.ET32OAN.HSAET32 UACC(NONE)
PERMIT HSA.ET32OAN.HSAET32 CLASS(FACILITY) ID(stcuser) ACC(READ)
```

With the SETROPTS command, the RACF class FACILITY is made available. With the SETROPTS RACLIST command, the FACILITY class resource profile copy in the RACF data space is enabled to increase performance. The next command, RDEFINE, fully qualifies the HSAET32 resource and sets an universal access of ″NONE″. With the PERMIT command, the RACF defined user stcuser gets READ access to this resource. Userid stcuser must be the user ID associated with your

NVSS started task. Note that with RACF you may specify the resource more generically if that is suitable for your environment.

Access to the CPCs

Each CPC in your AOFCUST definition must have a corresponding resource profile defined with your SAF product. The skeleton of the CPC resource is:

```
HSA.ET32TGT.netid.nau
```

The netid.nau part of the resource name corresponds with the netid.nau definition of the CPC entry in your AOFCUST definition. The period between netid and nau is part of the resource name. 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 with RACF you may specify the resource more generically 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
Retrieves, gets configuration information from the CPC

WRITE
Updates, sets configuration information of the CPC

CONTROL
Performs Operations Management Commands of the CPC

Note that this access level scheme is for the CPC and its LPAR levels.

Defining the CPC access level

Depending on the NVSS operator security chosen, the access level is checked differently. If your NVSS operator security (OPERSEC) is set to MINIMAL, NETVPW, or SAFPW, the userid that is checked for hardware access is always the userid that started the NVSS address space, which is usually an STC userid.

This userid has to be authorized for all CPC and CPC.Lpar resources you want to manage with this NetView. If multiple users are allowed to start the NVSS address space, make sure they are all authorized.

If you have chosen an NVSS operator security level of OPERSEC=SAFDEF or OPERSEC=SAFCHECK, the following paragraph applies.

With msys for Operations, individual NVSS users and NVSS autotasks need to be authorized to access the CPCs that are defined in your AOFCUST file. The current HSAET32 interface allows the user to perform the HW functions indirectly. No direct command interface is provided allowing the user to issue security level dependant commands. For this reason, the highest possible access level CONTROL has to be given to all NVSS users entitled to perform CF enable and CF drain functions for a specific CPC. In addition, the following NVSS autotasks need to be authorized with access level CONTROL for all defined CPCs:

- The XCF autotasks

- The autotasks defined with SYN %AOFOPXCFOPER% in automation table member AOFMSGA0
- AUTRCP
- AUTPLEX
- AUTBASE
- AUTHW*xxx*

The autotasks used for the HW interface initialization and communication also need to be authorized. Use access level CONTROL for the AUTHW*xxx* autotasks in your environment.

The AUTXCF*xx* autotasks and the additional ones from %AOFOPXCFOPER% are used internally if INGCF DRAIN or INGCF ENABLE is invoked by an authorized user. IXC102A message automation is also performed by these autotasks. The following example shows how to permit access to a CPC resource in RACF.

```
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A.

LPAR access example:

```
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A.* CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A and all its defined logical partitions.

Controlling access to the processor hardware functions

Chapter 5. Activating msys for Operations

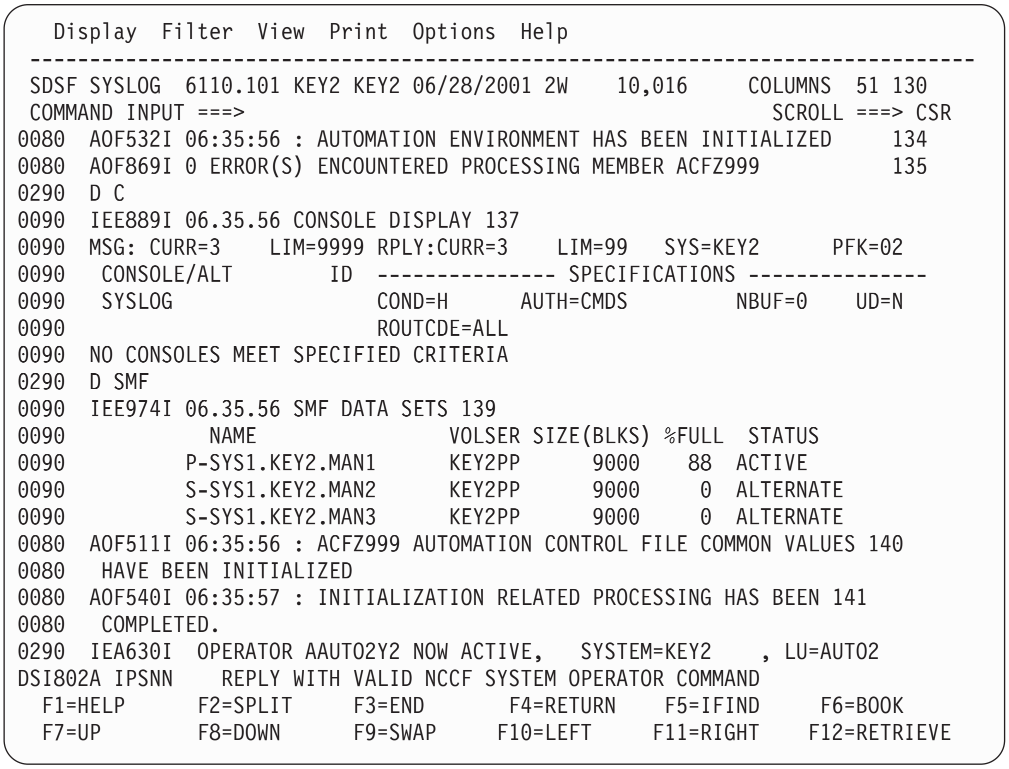
If you have completed all of the steps in the previous chapters, you are ready to start the msys for Operations program.

Starting msys for Operations using the MSOAPROC startup procedure

Enter the following at the system console:

```
S MSOAPROC
```

You will see messages similar to those in Figure 4.



```
   Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG  6110.101 KEY2 KEY2 06/28/2001 2W    10,016     COLUMNS  51 130
 COMMAND INPUT ===>                                            SCROLL ===> CSR
0080  AOF532I 06:35:56 : AUTOMATION ENVIRONMENT HAS BEEN INITIALIZED     134
0080  AOF869I 0 ERROR(S) ENCOUNTERED PROCESSING MEMBER ACFZ999           135
0290  D C
0090  IEE889I 06.35.56 CONSOLE DISPLAY 137
0090  MSG: CURR=3    LIM=9999 RPLY:CURR=3    LIM=99   SYS=KEY2      PFK=02
0090   CONSOLE/ALT        ID  --------------- SPECIFICATIONS ---------------
0090   SYSLOG                 COND=H      AUTH=CMDS         NBUF=0    UD=N
0090                          ROUTCDE=ALL
0090  NO CONSOLES MEET SPECIFIED CRITERIA
0290  D SMF
0090  IEE974I 06.35.56 SMF DATA SETS 139
0090            NAME                VOLSER SIZE(BLKS) %FULL  STATUS
0090          P-SYS1.KEY2.MAN1      KEY2PP      9000    88  ACTIVE
0090          S-SYS1.KEY2.MAN2      KEY2PP      9000     0  ALTERNATE
0090          S-SYS1.KEY2.MAN3      KEY2PP      9000     0  ALTERNATE
0080  AOF511I 06:35:56 : ACFZ999 AUTOMATION CONTROL FILE COMMON VALUES 140
0080   HAVE BEEN INITIALIZED
0080  AOF540I 06:35:57 : INITIALIZATION RELATED PROCESSING HAS BEEN 141
0080   COMPLETED.
0290  IEA630I  OPERATOR AAUTO2Y2 NOW ACTIVE,   SYSTEM=KEY2    , LU=AUTO2
DSI802A IPSNN    REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 4. Message output after starting msys for Operations

The important messages are DSI802A, AOF532I, and AOF540I.

The WTOR message DSI802A appears as follows:

```
*0003 DSI802A CNM01    REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND
```

This message indicates that NVSS has been started properly. You need not reply to this message as long as msys for Operations is supposed to run.

When you want to shut down msys for Operations reply `CLOSE IMMED` to DSI802A. Note that you must be authorized to issue that reply.

The other two messages (AOF532I, and AOF540I) look similar to the following:

```
AOF532I hh:mm:ss AUTOMATION ENVIRONMENT HAS BEEN INITIALIZED
AOF540I hh:mm:ss INITIALIZATION RELATED PROCESSING HAS BEEN COMPLETED
```

These messages indicate that initialization of the Sysplex Functions part has been successful.

Logging on to msys for Operations

To log on to msys for Operations, enter:

```
LOGON APPLID(domain_name) LOGMODE(logmode)
```

A logon panel is displayed, where you must specify one of the operator IDs defined before (see “Defining operators, passwords, and logon attributes” on page 75), and the appropriate initial password.

When your input has been accepted, you must change the password. Then the main menu of msys for Operations is be displayed. For details concerning the logon procedure, see Chapter 7, “Logging on to msys for Operations,” on page 103.

Chapter 6. Configuring msys for Operations for your environment

The following sections explain how to configure certain functions of msys for Operations for your environment.

Defining passwords for VSAM databases

You can define security passwords for the VSAM databases that are used by certain NVSS tasks. The general procedure for this is as follows:

1. Stop the task.
2. Modify the definition statements in INGALLC0 that define the database, changing them to include the specification of VSAM cluster passwords. Rerun job INGALLC0 using these modified statements to delete and redefine the database.
3. Update the initialization member in DSIPARM that is associated with the task by specifying the password for the password parameters.
4. Restart the task.

The following table lists the VSAM clusters for which this is possible:

Table 6. VSAM clusters for which passwords can be defined

VSAM cluster	Task	DSIPARM member	Parameter
NETVIEW.VnRnMn.*xxxxx*.DSILOGP	DSILOG	DSILOGBK	PPASS
NETVIEW.VnRnMn.*xxxxx*.DSILOGS	DSILOG	DSILOGBK	SPASS
NETVIEW.VnRnMn.*xxxxx*.DSISVRT	DSISVRT	DSISVRTD	PPASS

Printing the network log and trace log

The member CNMSJM04 in NETVIEW.VnRnMn..V1R4M0.CNMSAMP contains a sample print job for these logs. Copy this member to SYS1.PROCLIB and rename it to CNMPRT. If you defined passwords for the network log and the trace log, add a password statement to job CNMPRT.

To change the defaults used to print the network or trace logs, control statements must be passed to PGM=DSIPRT using the DSIINP DD statement. You can do this using one of two methods:

1. Create the following statements for a job stream or an in-stream procedure:

   ```
   //DSIINP   DD *
            PASSWD=password
            OPER1,OPER2,NETOP1
            TRANSTBL MOD=DSIEBCDC
   ```

2. Create a statement similar to the following to define a data set member to contain the print control statements and put the preceding print control statements in this member.

   ```
   //DSIINP  DD  DSN=SYS1.PARMLIB(MEMBER),DISP=SHR
   ```

Only the second method applies for system-started JCL procedures.

Part 3. Using the msys for Operations operator interface

This part describes how to use the msys for Operations operator interface, and has the following chapters:

Chapter 7. Logging on to msys for Operations

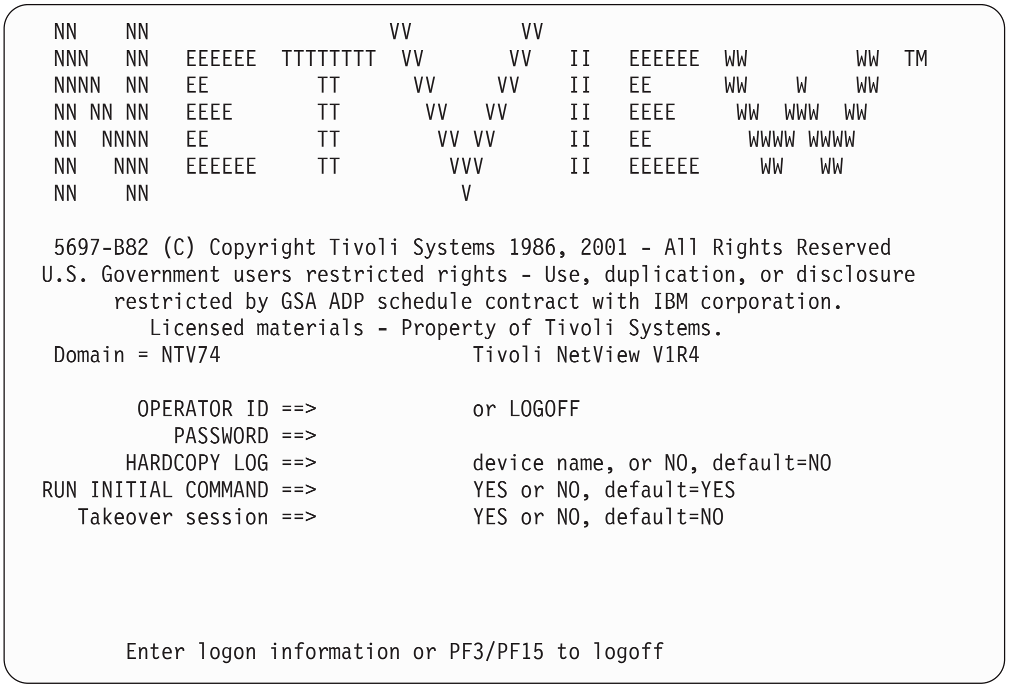
1. To log on to msys for Operations, enter:

   `logon applid(`*`applid`*`) logmode(`*`logmode`*`)`

   where *applid* is the domain name of the msys for Operations application to which you are logging on. LOGMODE is an optional parameter that specifies information about your terminal session.

   When you log on, msys for Operations queries the device for screen size and color attributes if the logmode specifies to issue the query. Otherwise, msys for Operations uses the screen size specified in the logmode. The command facility adapts to use the entire width and depth of the screen. All components of the msys for Operations program support color where the display is capable of displaying color.

   When a session is established, a msys for Operations logon panel similar to the one shown in Figure 5 is displayed.



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

 5697-B82 (C) Copyright Tivoli Systems 1986, 2001 - All Rights Reserved
U.S. Government users restricted rights - Use, duplication, or disclosure
      restricted by GSA ADP schedule contract with IBM corporation.
         Licensed materials - Property of Tivoli Systems.
 Domain = NTV74                     Tivoli NetView V1R4

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




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

Figure 5. Example of msys for Operations logon panel

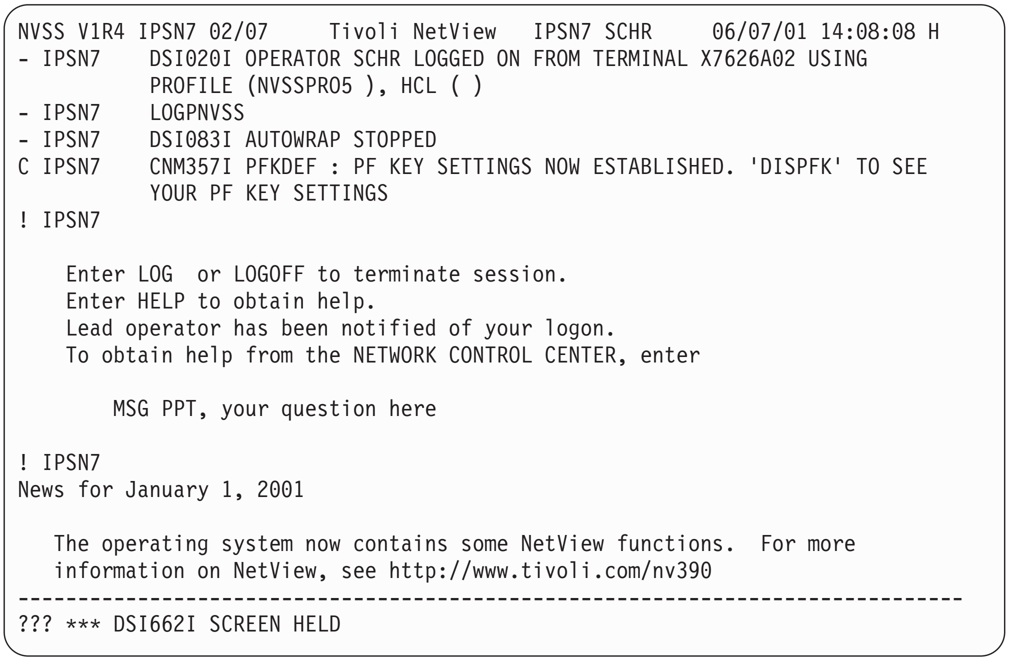
2. Enter your operator identification (for example, OPER1) in the space next to the `OPERATOR ID` field, where the cursor is located.

   Blanks entered in the msys for Operations logon fields will be treated as null characters. For example, OPER 1 entered in the OPERATOR ID field of the msys for Operations logon screen will be treated as OPER1 because the blank between "R" and "1" is treated as a null character.

3. Enter your password. If you want to change your password, leave this field blank.

4. If you do not want to use an initial command, enter `no` in the `RUN INITIAL COMMAND` field. If you want to use an initial command, leave this field blank or enter `yes`. The initial command is set up by your system programmer to eliminate some manual procedures.

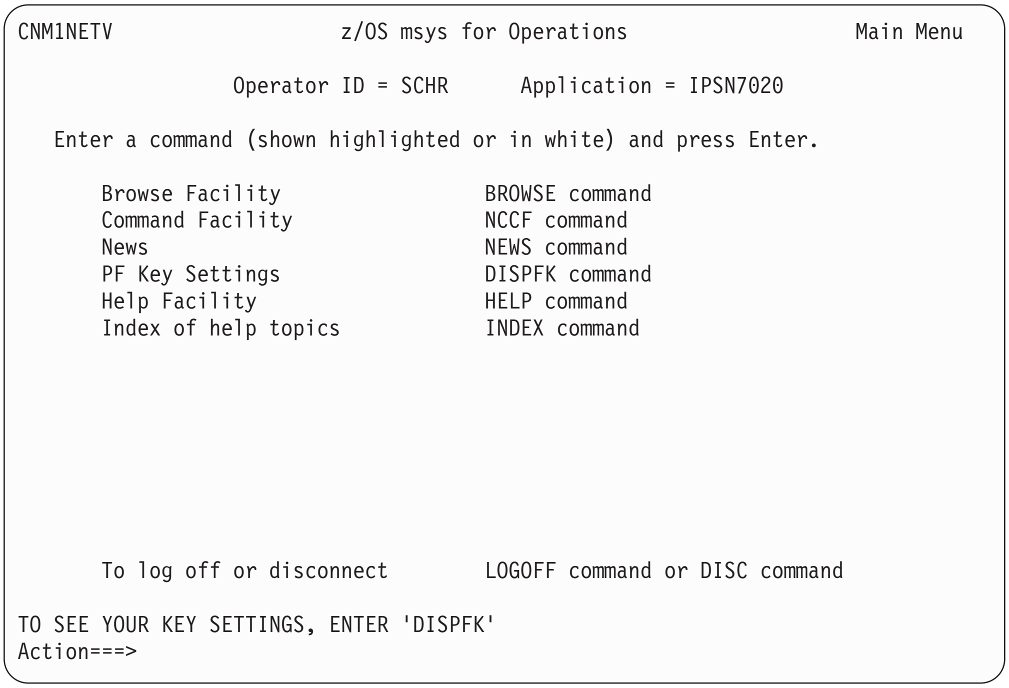
5. Press Enter. A panel similar to Figure 6 is displayed.



```
NVSS V1R4 IPSN7 02/07     Tivoli NetView   IPSN7 SCHR     06/07/01 14:08:08 H
- IPSN7    DSI020I OPERATOR SCHR LOGGED ON FROM TERMINAL X7626A02 USING
           PROFILE (NVSSPRO5 ), HCL ( )
- IPSN7    LOGPNVSS
- IPSN7    DSI083I AUTOWRAP STOPPED
C IPSN7    CNM357I PFKDEF : PF KEY SETTINGS NOW ESTABLISHED. 'DISPFK' TO SEE
           YOUR PF KEY SETTINGS
! IPSN7

    Enter LOG  or LOGOFF to terminate session.
    Enter HELP to obtain help.
    Lead operator has been notified of your logon.
    To obtain help from the NETWORK CONTROL CENTER, enter

        MSG PPT, your question here

! IPSN7
News for January 1, 2001

   The operating system now contains some NetView functions.  For more
   information on NetView, see http://www.tivoli.com/nv390
-------------------------------------------------------------------------------
??? *** DSI662I SCREEN HELD
```

Figure 6. msys for Operations news panel

6. Press the Clear or Enter key to go to the msys for Operations Main Menu. Possibly, you must do this more than once. After msys for Operations processes the operator profile, the main menu is displayed:



```
CNM1NETV                   z/OS msys for Operations                   Main Menu

                  Operator ID = SCHR      Application = IPSN7020

   Enter a command (shown highlighted or in white) and press Enter.

       Browse Facility                 BROWSE command
       Command Facility                NCCF command
       News                            NEWS command
       PF Key Settings                 DISPFK command
       Help Facility                   HELP command
       Index of help topics            INDEX command









       To log off or disconnect        LOGOFF command or DISC command

TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
Action===>
```

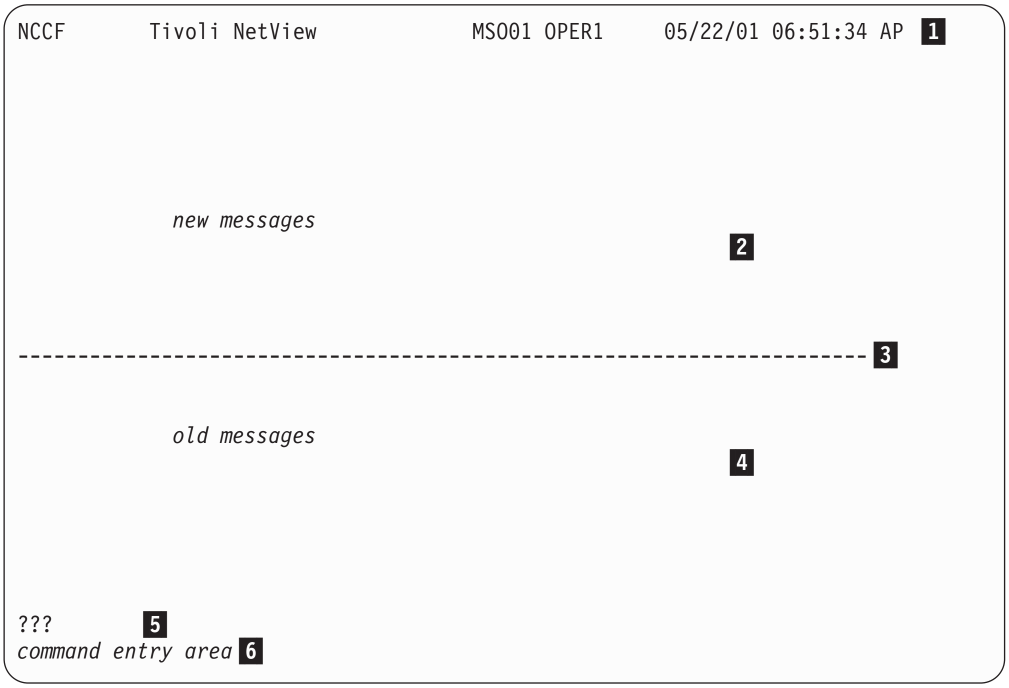
Figure 7. msys for Operations main menu

7. The main menu contains a command line from which you can issue commands to msys for Operations.

Chapter 8. The message display screen

To access the message display screen, enter `nccf` on the main menu panel and press Enter. You can also use the ROLL function (PF6 by default). For further information about the ROLL function refer to "Moving between components" on page 111. In order to return to the main menu, enter `mainmenu` in the command entry area or use the ROLL function again.

The layout of the message display screen is as follows:



```
NCCF       Tivoli NetView            MSO01 OPER1     05/22/01 06:51:34 AP

            new messages

---------------------------------------------------------------------

            old messages

???
command entry area
```

Figure 8. Sample message display screen

Session identification line

The first line of the panel, identified with 1, gives you the name of the panel that appears. The next field lists the domain name (MSO01) and your operator identifier (OPER1). The next two fields list the current date and time. The last two fields contain a combination of A, H, P, W, or a blank, which indicates whether messages can be written to the panel. The A, H, P, and W indicators are described in the following list:

A The autowrap indicator means that AUTOWRAP is active. If autowrap is on and the display is full of data, it is automatically overlaid with new data. If autowrap is not on, press the Clear or Enter key to allow new data to overlay the display screen.

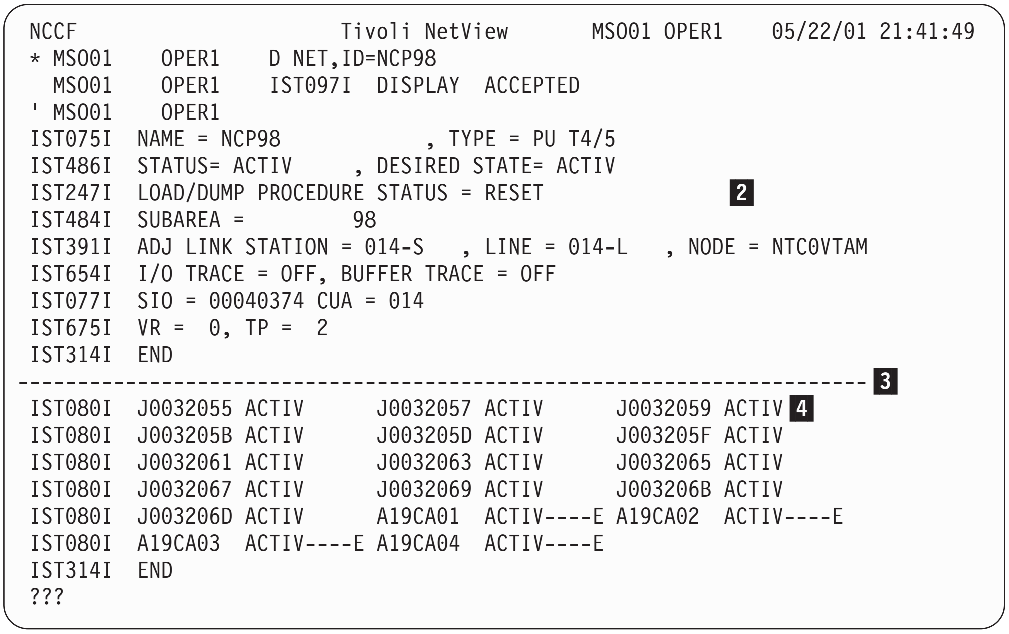
H The held-screen indicator means that the screen does not roll forward unless it is unlocked by the operator. You can use this indicator if you need time to read the screen before it is erased, or to freeze the screen while you mark messages for deletion or enter a command.

P The pause status indicator. A command list running on the operator task is pausing for operator input, and will not continue until the operator enters information.

W The wait indicator. A command list running on the operator task is waiting for messages or other events, such as for a specified amount of time to elapse.

Message area

The message area displays commands, responses, and messages from the system. Figure 9 shows a sample display screen.



```
 NCCF                       Tivoli NetView       MS001 OPER1    05/22/01 21:41:49
 * MS001    OPER1    D NET,ID=NCP98
   MS001    OPER1    IST097I  DISPLAY  ACCEPTED
 ' MS001    OPER1
 IST075I  NAME = NCP98            , TYPE = PU T4/5
 IST486I  STATUS= ACTIV     , DESIRED STATE= ACTIV
 IST247I  LOAD/DUMP PROCEDURE STATUS = RESET                2
 IST484I  SUBAREA =         98
 IST391I  ADJ LINK STATION = 014-S   , LINE = 014-L   , NODE = NTC0VTAM
 IST654I  I/O TRACE = OFF, BUFFER TRACE = OFF
 IST077I  SIO = 00040374 CUA = 014
 IST675I  VR =  0, TP =  2
 IST314I  END
----------------------------------------------------------------------- 3
 IST080I  J0032055 ACTIV      J0032057 ACTIV      J0032059 ACTIV 4
 IST080I  J003205B ACTIV      J003205D ACTIV      J003205F ACTIV
 IST080I  J0032061 ACTIV      J0032063 ACTIV      J0032065 ACTIV
 IST080I  J0032067 ACTIV      J0032069 ACTIV      J003206B ACTIV
 IST080I  J003206D ACTIV      A19CA01  ACTIV----E A19CA02  ACTIV----E
 IST080I  A19CA03  ACTIV----E A19CA04  ACTIV----E
 IST314I  END
 ???
```

Figure 9. Sample display screen

The dashed line, indicated by 3 separates the latest messages from the older ones. The messages are continually updated. You can use this line to locate the most recent messages. The most recent message is the one directly above the line, in the area indicated by 2 . The older messages displayed on the screen are below the line, in the area indicated by 4 . The oldest message is the one directly below the line.

The first line of the sample screen indicates that the VTAM command `D NET,ID=NCP98` was issued by OPER1 on domain MSO01. The second line indicates that VTAM has accepted the command. The rest of the upper part of the screen consists of the output of the DISPLAY command. The codes in the first column of the first three lines indicate the type of the respective message. For the meaning of the type codes, see Table 8 on page 114.

Generally, messages disappear as the screen scrolls. Examples of exceptions include reply messages, held messages, and windowed responses.

Reply messages

Reply messages are messages to which you should reply before you delete them from the display screen. These messages appear in high intensity on your display screen with a reply number. Unsolicited reply messages received on the system console remain outstanding even after a reply is given. Delete these messages manually using the MVS control (K) command.

Held messages

Held messages are messages that are defined to be held on the screen. These messages appear in high intensity (or are otherwise highlighted) and appear at the top of the message area. Specific action must be taken to remove them, such as:

- De-emphasizing them with a Delete Operator Messages (DOM) command
- Specifically deleting them (by the operator)

The DOM command causes messages to lose highlighting immediately. This means they can now scroll off the screen. If there are more messages being held than can be displayed on your type of terminal, message DSI151I appears and the messages are queued. The queued messages appear only when existing ones are deleted.

To delete one or more held messages:

1. Move the cursor to the message line, using either the cursor keys or the TAB key.
2. To delete a single message, press **Enter**. The cursor will return to the command entry area.
3. To delete multiple messages, erase the first line of each message to be deleted (you can use the Erase EOF key) and press **Enter**. The cursor will return to the command entry area.

Attention: If an autowrap timeout occurs while you are typing over message text, that text might be moved or refreshed, thus destroying the typing that has been done.

To avoid losing information from the command entry area, you can either:

- Turn autowrap off, using the AUTOWRAP NO command.
- Use the HOLD command.

Windowed responses

Windowed responses are messages that are displayed in a scrollable window using the msys for Operations WINDOW command. This prevents the message responses from being overwritten by subsequent messages, and enables you to navigate through the information using standard BROWSE commands. For a description of the behavior of windowed responses, refer to the WINDOW command in the msys for Operations online help.

Response area

Near the bottom of the screen is a line that begins with the ??? indicator. This line is the response area, indicated by **5** in Figure 8 on page 105. Look here for error messages.

The =X= indicator is displayed in place of the ??? indicator when messages are arriving (prior to entering or after leaving a panel). This indicator means that only a limited set of commands can be used. Some of the commands you can use are:

- AUTOWRAP
- LOGOFF

Command entry area

The cursor is located in the command entry area, indicated by 6 in Figure 8 on page 105. You communicate with the msys for Operations program by entering commands here or you can invoke another msys for Operations component (for example, by entering `mainmenu`).

Chapter 9. Issuing commands

You can issue commands from the message display screen or from the menu. If you press a key on a terminal that has no keyboard buffering capability, and the controller is already processing a request from the host, the key is rejected, and the keyboard can lock up. You can then press **RESET** to unlock the keyboard and enable input to proceed.

The length of the command entry area is limited to three lines of 80 characters each. For input modes of two or three lines, on screens wider than 80 characters, the msys for Operations program indicates the end of the input area with three less-than symbols (<<<). When you press any action key (Enter, any PF or PA key, or Clear), the command area is erased.

Repeating commands

The RETRIEVE command tells the system to place the last command you entered on the command line. If necessary, you can alter the command on the command line, or leave it as it is, then press Enter to send the command to the system.

You can repeat the RETRIEVE command several times to display the last few commands that you sent to the system. The easiest way to use the RETRIEVE command is by assigning it to a PF key. The msys for Operations-supplied default for the RETRIEVE command is PF12.

Issuing MVS system commands

To issue commands to MVS, use the MVS command, which enables you to control MVS system operations without using a separate screen for multiple console support (MCS).

To issue a command from the msys for Operations command facility, enter `MVS` followed by a valid MVS command. For example, to display a list of active MVS tasks, enter:

```
mvs d a,l
```

msys for Operations displays the response from MVS on the message display screen.

Chapter 10. Moving between the components and using function keys

This chapter describes how you can move between the components of msys for Operations, and how to use and display function keys.

Moving between components

The msys for Operations program allows you to have multiple components active at the same time. For example, the message display screen (which is always active), the main menu, and a output panel of the BROWSE command can be active simultaneously. One method to move from one component to another is to enter the component name. Thus, you can enter `nccf` on the command line of the BROWSE panel to move to the message display panel.

You can also use the ROLL function to move among active components in a continuous loop. The PF key for ROLL that is supplied by msys for Operations is PF6. If your PF key settings have PF6 set to ROLL, then pressing PF6 returns you to the last panel you viewed in an active component.

Using program function keys

You can use program function (PF) or program access (PA) keys to send commands to the system. Doing so can save time because you do not have to type a command and then press the Enter key.

Note: Occasionally, using PF keys may lead to unexpected results. This is because PF keys work such that they do not accept new values in input fields. Instead, they reset the input fields to the values last entered with the Enter key and then run the requested function.

Listing PF and PA keys

Most PF and PA keys have already been set for you, with unique settings by component. They are set to commands that you will use quite often.

To display the current settings for the message display screen PF and PA keys, enter:

```
dispfk nccf
```

You can also display PF key settings for other components, such as log browse, by specifying their component abbreviations on the DISPFK command or a PF key set to that command. For example, the msys for Operations defaults specify the DISPFK command with the APPEND keyword as PF4, allowing you to enter a component name on the command line, then press PF4 to see that component's PF keys. Browse the CNMKEYS member or enter `dispfk all` to display all PF key settings.

You can change the key settings for your actual session with the SET command; see "SET" on page 129.

Chapter 11. Using the netlog

The netlog is the record of the terminal activity that has occurred on the system. You can send commands, responses, and messages to the netlog. Each message contains the time and date it was sent and the names of the operator and domain it came from.

The netlog is stored in two VSAM data sets. One of these is the primary (referred to as `netlogp`), and the other the secondary log data set (referred to as `netlogs`). Only one log data set is used at any point in time. This is the active log data set (referred to as `netloga`). The other one is inactive (referred to as `netlogi`). At first, the primary data set is active, and the secondary inactive. When the primary log data set becomes full, logging is continued on the secondary log data set. Now the secondary data set is active, and the primary is inactive. When the secondary data set becomes also full, logging switches back to the primary data set; the old entries will be overwritten.

You can print the inactive netlog file in batch mode, while the system is using the active file as the log.

Displaying the netlog

You can use the BROWSE command to display a particular netlog data set. You can select the active or inactive log, or you can name the specific log (primary or secondary) to browse. For example, to display the active log, enter:

```
browse netloga
```

If the primary log data set is the active one, the command

```
browse netlogp
```

has the same effect.

You can specify a time and date range to limit the amount of netlog information displayed. For example, to display the primary netlog from 1:00 p.m. on 4/07/01 to 8:30 a.m. on 4/08/01, enter:

```
browse netlogp from 4/07/01 13:00 to 4/08/01 8:30
```

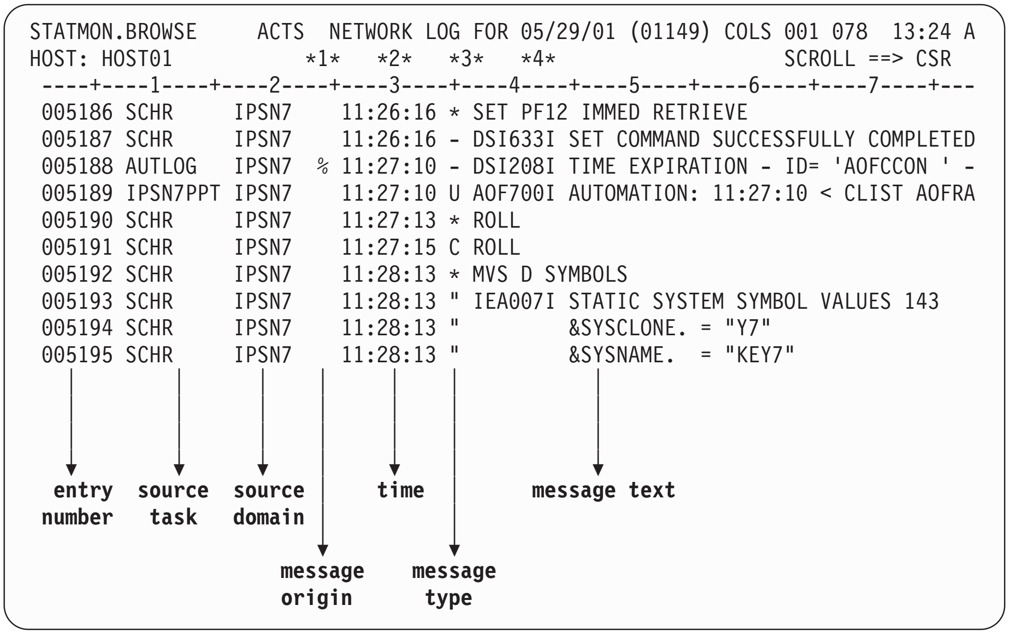
Note: If you specify a time range for browsing the netlog, the first and the last record of the specified time range remains the first and the last record during the entire browse.

You can use the FIND or ALL commands to locate specific information while you are browsing the netlog. For example, to find the words INVALID COMMAND, enter:

```
f 'invalid command'
```

Structure of the log entries

The following figure shows a netlog panel with a short explanation of the meaning of the individual columns:



```
STATMON.BROWSE     ACTS  NETWORK LOG FOR 05/29/01 (01149) COLS 001 078  13:24 A
HOST: HOST01           *1*   *2*   *3*   *4*                  SCROLL ==> CSR
 ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
 005186 SCHR     IPSN7    11:26:16 * SET PF12 IMMED RETRIEVE
 005187 SCHR     IPSN7    11:26:16 - DSI633I SET COMMAND SUCCESSFULLY COMPLETED
 005188 AUTLOG   IPSN7  % 11:27:10 - DSI208I TIME EXPIRATION - ID= 'AOFCCON ' -
 005189 IPSN7PPT IPSN7    11:27:10 U AOF700I AUTOMATION: 11:27:10 < CLIST AOFRA
 005190 SCHR     IPSN7    11:27:13 * ROLL
 005191 SCHR     IPSN7    11:27:15 C ROLL
 005192 SCHR     IPSN7    11:28:13 * MVS D SYMBOLS
 005193 SCHR     IPSN7    11:28:13 " IEA007I STATIC SYSTEM SYMBOL VALUES 143
 005194 SCHR     IPSN7    11:28:13 "         &SYSCLONE. = "Y7"
 005195 SCHR     IPSN7    11:28:13 "         &SYSNAME.  = "KEY7"
```

Figure 10. Network log entries

The following table explains the codes that can occur in the **Message Origin** column.

Table 7. Message origin codes

Code	Explanation
P	Message from the PPT task (VTAM)
%	Message to the authorized message receiver
P%	Message to the authorized receiver from the PPT
*	Message to a secondary receiver
P*	Message to a secondary receiver from the PPT
+	Message to a copied receiver (assigned with COPY)

The following table explains the more important codes that can occur in the **Message Type** column.

Table 8. Message type codes

Code	Explanation
C	Message or command generated by a CLIST
E	Message from the operating system interface
M	Message from a message command
Q	Unsolicited message from VTAM
R	Indicates that an operator entered the VTAM REPLY command in response to NetView WTOR number DSI802A. This message type is logged but does not appear on msys for Operations consoles.
S	Message text modified by user exit
t	Indicates a message response from a TSO command.
U	Message from locally-written programming
V	VTAM command entered from the system console

Table 8. Message type codes (continued)

Code	Explanation
X	Indicates a cross-domain (NNT-OST) command.
Y	VTAM message from the system console.
Z	Indicates a message from a data services task (DST).
!	Indicates a message from an immediate command processor. When displayed in the immediate message area on the screen, the message type and domain name are not displayed. When received cross-domain, this type of message is in the normal output area, along with its domain name and type prefix.
–	Message from NVSS
*	Command from a terminal operator (command echo)
+	Message from programs other than msys for Operations
>	Message requiring a reply
′	Multi-line message from msys for Operations
″	Multi-line message from MVS
=	Mult-line message from non-IBM code
|	Indicates a message generated in a pipeline.

Log browse filtering

The BLOG command activates the netlog browse facility based on filters. You can select which records to display using any combination of the following filters:

- Select a local or remote msys for Operations. The default is the local msys for Operations. Changing the msys for Operations domain, Netid, or operid fields may result in browsing a remote msys for Operations log.
- Select the NETLOGA, NETLOGI, NETLOGS, or NETLOGP log.
- Select the starting display column.
- Select the operator ID for which records were logged.
- Select the origin domain of records that were logged.
- Select the message ID of messages that were logged.
- Select the starting time and date for records that were logged.
- Select the ending time and date for records that were logged.
- Select a character string that will be matched with the text of a message that was logged.

Chapter 12. Getting online help

To access the msys for Operations online help, enter `HELP` *`item`* on the command line, where *item* is the item for which you want help. This can be a command, a message ID, or an abend code. For the sysplex commands INGCF and INGPLEX, you can also specify the subcommand you are interested in. If you enter `HELP` without any additional parameters, a help menu for NVSS is displayed; note, however, that the INGCF and INGPLEX commands are not included in this menu.

Examples

To get help for the DRAIN subcommand of the Sysplex Functions command INGCF, enter:

```
HELP INGCF DRAIN
```

To get help for the NVSS BROWSE command, enter:

```
HELP BROWSE
```

To get help for message CNM937I, enter:

```
HELP CNM937I
```

To get help for a user abend code (U*xxx*), enter:

```
HELP ABEND
```

Note: If you receive an msys for Operations message that includes an unexplained return code, see “Macro return codes” on page 255.

Getting online help

Part 4. Command reference

This part provides a reference for the commands available in msys for Operations, and has the following chapters:

Chapter 13. General commands

This chapter describes the commands that are supplied by NVSS. For online help on a specific command, enter:

```
HELP command
```

where *command* is the name of the command.

ALL (BROWSE, WINDOW)

Purpose

Use the ALL command to display a specified collection of lines in BROWSE and WINDOW. If parameters are not specified, all lines are displayed and current filtering is disregarded. If */string/* is specified, only lines matching the string are displayed.

For more information, refer to the online help.

Examples

Displaying only lines containing characters TASK

To display only lines with the characters TASK, enter the following command from the BROWSE or WINDOW command line:

```
ALL /TASK/
```

AUTOWRAP

Purpose

The AUTOWRAP command controls whether your terminal is held when the screen is full of data, or if the screen is automatically overlaid with new data.

For more information, refer to the online help.

Examples

Setting wrap display time

To set AUTOWRAP to display new data seven seconds after the screen is full, enter:

```
AUTOWRAP 7
```

BACK (BROWSE, HELP, WINDOW)

Purpose

The BACK command scrolls backward to the beginning of the data.

For more information, refer to the online help.

Examples

Displaying a help panel further than one panel back

If you want to navigate to a help panel that is three pages back, enter one of the following commands:

```
BACK 3
```

```
B 3
```

BLOG

Purpose

The BLOG command activates the log browse facility, showing a subset of the information based on filtering criteria.

When used with no parameters, the BLOG command will display a full-screen input panel where the filtering criteria can be entered. When used with parameters, the BLOG command will start the log browse facility based on the filtering criteria provided on the command line arguments.

For more information, refer to the online help.

Examples

Using the BLOG input panel

To use the BLOG input panel, enter:

```
    BLOG
```

The BLOG input panel is displayed where you can enter filtering information. When you have entered your choices, press **Enter** to start the log browse facility.

BOTTOM (BROWSE, HELP)

Purpose

The BOTTOM command displays the last page of a multipage panel.

For more information, refer to the online help.

BROWSE

Purpose

The BROWSE command enables you to scan the netlog or members of a partitioned data set (PDS).

The members are contained in a partitioned data set.

For more information, refer to the online help.

Examples

The format of dates and times specified in the following examples assumes the default setting for date and time formats on the DEFAULTS and OVERRIDE commands.

Browsing the active netlog for a specified time

To browse the netlog (either primary or secondary) that is currently active from 2:40 p.m. on February 5, 2001 to 2:00 a.m. on February 6, 2001, enter:

```
BROWSE NETLOGA FROM 02/05/01 14:40 TO 02/06/01 2:00
```

Browsing the inactive netlog

To browse the netlog (either primary or secondary) that is currently inactive, enter:

```
BROWSE NETLOGI
```

Browsing a DSICMD member

When you want to browse a DSICMD member, but do not wish to have the included members resolved, enter:

```
BROWSE DSICMD NOINCL
```

DEFAULTS

Purpose

The DEFAULTS command sets msys for Operations-wide defaults.

You can override some of the DEFAULTS command settings for a specific operator ID using the OVERRIDE command.

You can use the LIST DEFAULTS command to get a list of the current DEFAULTS settings and the number of dumps that have been taken for storage overlay or control block overwrite conditions (DMPTAKEN).

For more information, refer to the online help.

Examples

Changing the banner

To change the banner on the logon and message display panels, enter one of the following:

```
DEFAULTS BANNER=OneBigWord
```

```
DEFAULTS BANNER='Up to 24 characters'
```

DISPFK

Purpose

The DISPFK command enables you to display or save the PF key settings.

For more information, refer to the online help.

END (BROWSE, HELP)

Purpose

The END command stops the current component panel sequence and returns to the component that was previously active.

For more information, refer to the online help.

FIND (BROWSE)

Purpose

The FIND command locates specific information while browsing a data set and a member. You can search for a previous entry or for the next entry. The default is NEXT. You can limit columns to be searched by specifying *left* and *right* column numbers.

For more information, refer to the online help.

Examples

Finding the next occurrence of a specified string

To find the next occurrence of DSI, enter:

```
FIND DSI
```

or

```
F DSI
```

Finding the previous occurrence of a specified string

To scan the lines previous to the current line for an occurrence of the string DSIDMN, enter:

```
FIND DSIDMN PREV
```

or

```
F DSIDMN P
```

FIND (WINDOW)

Purpose

The FIND command locates specific information while displaying data with the WINDOW command. This includes command and message help, and index information. You can search for a previous entry or for the next entry. The default is NEXT. You can limit columns to be searched by specifying *left* and *right* column numbers.

The search begins where the cursor is located, if the cursor is in the display. Otherwise, the search begins at the first line of information displayed on your screen.

For more information, refer to the online help.

Examples

Finding the next occurrence of a specified string that limits the search to specified columns

To find the next occurrence of DSIDMN in columns 1 to 90, enter:

```
F 'DSIDMN' 1 90
```

FORWARD

Purpose

The FORWARD command scrolls forward toward the end of the data.

For more information, refer to the online help.

Examples

Advancing a specified number of help panels

If you want to move ahead five help panels, enter:

```
FORWARD 5
```

HELP

Purpose

The HELP command displays help information for messages and commands.

You can use the following commands while you are using the HELP facility:
- BACK
- BOTTOM
- END
- FORWARD
- HELP
- RETURN
- TOP

For more information, refer to the online help.

Examples

Displaying help for commands

To receive help for the INGPLEX command, enter:

```
HELP INGPLEX
```

Displaying an online help panel for a specified message

To display the online help panel for message CNM937I, enter:

```
HELP CNM937I
```

The action suffix (I) is not required.

INDEX

Purpose

The INDEX command displays topics that are explained in the online help facility. Use the backward and forward PF keys to move through the index. Use the FIND command to search for a particular topic. You can select a highlighted help selection either by typing the code (indicated in the right-hand column) on the command line or by placing the cursor on the selection (you can tab to this line). To view the help topic, press Enter.

For more information, refer to the online help.

Examples

Displaying index entries

To display the online index, enter:

```
INDEX
```

To display all index entries beginning with the letter R, enter:

```
INDEX R
```

INPUT

Purpose

The INPUT command modifies the length of the input area of the message display screen. The input area is at the bottom of the message display screen.

For more information, refer to the online help.

Examples

Changing the input area to a specified number of lines

To change the command entry area to two lines, enter:

```
INPUT 2
```

LIST

Purpose

The LIST command gives information about your msys for Operations session.

For more information, refer to the online help.

Examples

Displaying the current defaults

To display the current defaults, enter:

```
LIST DEFAULTS
```

For an explanation of the various defaults, see “DEFAULTS” on page 123. Note that some values (for example, SENDMSG, SCRNFMT, and SCROLL) are not valid on the DEFAULTS command.

LISTA

Purpose

The LISTA command displays the data set status, disposition, *ddnames*, and data set names of the files currently allocated to the msys for Operations program. It can also indicate which data sets contain a specific member.

The LISTA command lists the files allocated to the msys for Operations program. This includes files allocated through JCL and those allocated dynamically. In addition, (OPER-DS) indicates an operator data set designated by the OVERRIDE command. Also, (INSTORE-COMMON) indicates a member loaded by the INSTORE stage.

For more information, refer to the online help.

Examples

Listing a file with a specified ddname and member

The following example illustrates how to find the data set in DSIPARM that contains member DSITBL01:

```
LISTA DSIPARM DSITBL01
```

Listing all allocated files

To list all allocated files, enter:

```
LISTA
```

LOCATE (BROWSE)

Purpose

With the LOCATE command you can position your log browse display at a particular record number, or date and time. When used without parameters, the LOCATE command positions the log browse display at the first record on the date currently being displayed.

For more information, refer to the online help.

Examples

Locating 11 a.m. on today's date

To locate the log browse display at 11 a.m. on today's date, enter:

```
LOCATE 11:00 TODAY
```

LOGOFF

Purpose

The LOGOFF command ends the session between your terminal and the system. When your task terminates, some of your messages are rerouted to another authorized receiver. The messages rerouted include all those messages from VTAM or from your MVS operating system that require a reply or action from you, and any messages that were routed to you as a primary receiver but were not processed at the time of termination.

For more information, refer to the online help.

MSG

Purpose

The MSG command sends a message to an operator or to the netlog.

For more information, refer to the online help.

Examples

Sending a message to all active terminals and system console operators

To send a message indicating system shutdown to all active terminals and to the system console operator, enter:

```
MSG ALL,SYSTEM SHUTDOWN IN 15 MINUTES
```

MVS

Purpose

The MVS command enables you to enter an MVS system operator command from the msys for Operations program. If your task has not obtained an MVS console, the MVS command attempts to obtain one for you.

For more information, refer to the online help.

Examples

Using the MODIFY command

To use the MODIFY command, enter:

```
MVS MODIFY TSO,USERMAX=nnnn
```

OVERRIDE

Purpose

The OVERRIDE command can be used to specify options for a particular operator. The OVERRIDE options take precedence over the options specified by the DEFAULTS command. The OVERRIDE options that are related to message display (BEEP, DISPLAY, HOLD) apply to all messages that are to be displayed at the individual operator's terminal.

Use the LIST OVERRIDE command to request a list of the current OVERRIDE settings.

For more information, refer to the online help.

Examples

Changing the command priority

To change the command priority, enter:

```
OVERRIDE CMD=low
```

REPEAT (BROWSE)

Purpose

The REPEAT command reissues the last FIND command while you are browsing the netlog or a member of a partitioned data set. Because the REPEAT command is sensitive to the current position of the cursor, it is normally entered using a PF key.

By repeatedly pressing the PF key set to REPEAT, you can find successive occurrences of a specified character string. When the first occurrence of a character string has been found, the REPEAT key will find the next occurrence. When the last occurrence of a character string has been found, the REPEAT key can be used to continue the search, wrapping around from the bottom line to the top line (or from the top line to the bottom line if the FIND command included the PREV parameter).

For more information, refer to the online help.

RETRIEVE

Purpose

The RETRIEVE command places the last command you issued in the command input area. This command gives you a convenient method to review, rerun, or edit and rerun commands you have recently entered.

For more information, refer to the online help.

RETURN (BROWSE, HELP)

Purpose

The RETURN command returns you to the previous component or the last selection panel that you used.

For more information, refer to the online help.

ROLL

Purpose

The ROLL command returns to a previous component and the last panel that you used in that component.

The system remembers the sequence in which you go from one component to another. When you use the ROLL command, the system moves the name of your current component to the beginning of the sequence of components, and brings up the component at the end of the sequence, displaying the panel that was displayed when you left that component.

For more information, refer to the online help.

SET

Purpose

The SET command defines PA and PF keys for the command facility or a full-screen application that supports its PF or PA settings. These settings remain valid until you delete them or log off.

For more information, refer to the online help.

Note: The PF key settings for the INGPLEX and INGCF commands cannot be changed with the SET command.

Examples

Setting PF12 to retrieve your last command

To set the PF12 key to retrieve your last command, enter:

```
SET PF12 IMMED RETRIEVE
```

TOP (BROWSE, HELP)

Purpose

The TOP command displays the first page of a multipage panel.

For more information, refer to the online help.

WHO

Purpose

The WHO command list displays the status of all operator terminals and information about your session.

After entering the WHO command list, you see information similar to the following:

```
* MSO01    WHO
C MSO01    LIST STATUS=OPS
- MSO01    OPERATOR: OPER1    TERM: A01A701  STATUS: ACTIVE
- MSO01    OPERATOR: AUTO1    TERM: AUTO1    STATUS: ACTIVE
- MSO01    OPERATOR: AUTO2    TERM: AUTO2    STATUS: ACTIVE
- MSO01    END OF STATUS DISPLAY
C MSO01    LIST STATUS=NNT
- MSO01    MAX SESS: 00000005
- MSO01    NO ACTIVE NCCF TO NCCF SESSIONS FOUND
C MSO01    LIST OPER1
- MSO01    STATION: OPER1    TERM: A01A701
- MSO01    HCOPY: NOT ACTIVE PROFILE: DSIPROFA
- MSO01    STATUS: ACTIVE
- MSO01    AUTHRCVR: NO      CONTROL: GLOBAL
- MSO01    OP CLASS LIST: 2
- MSO01    DOMAIN LIST: MSO01 (I) MSO02 (I) MSO03 (I)
- MSO01    ACTIVE SPAN LIST: NONE
- MSO01    END OF STATUS DISPLAY
```

For more information, refer to the online help.

WINDOW

Purpose

The WINDOW command is a full–screen application that captures and displays data from other commands that would normally display messages. The WINDOW command facilitates searching the captured data and enables you to scroll forward and back, as well as left and right. WINDOW is also ROLLable.

For more information, refer to the online help.

Examples

Displaying a data set wider than 80 characters

To display member XYZ of data set USER.LISTING which is greater than 80 characters wide, enter:

```
WINDOW < 'USER.LISTING(XYZ)'
```

General commands

Chapter 14. Sysplex-related commands

This section contains reference information about the INGCF, INGCFL, INGHC and INGPLEX commands, which support several actions. Some of these actions impact the system configuration. Others only serve to display information. The display actions are accessible for every operator that can call the respective command. Access to related groups of actions that modify the system configuration can be granted or denied to operators individually.

Note: The actions that are controlled by this security mechanism are marked by an asterisk (’*’) in the following descriptions.

The layout of the panels from which you initiate an action depend on your authorizations. The code or PF key by which you initiate a certain action is only displayed when you are authorized to perform the action.

Additional parameters for System Operations commands

The following parameters are available for a number of system operations commands:

OUTMODE

This parameter lets you specify the output mode of a command. If you specify LINE, the output is displayed in line mode, independent of the task type.

The syntax is as follows:

```
►►──Command name──┬──────────────┬────────────────────────────────────►◄
                  └─OUTMODE=LINE─┘
```

Further characteristics are the following:

- No color attributes are set for data that is shown in line mode.
- The sequence of the fields may be different in line mode than in fullscreen.
- Not all fields from the fullscreen display may be shown in line mode.
- Line mode output is shown in a multiline message.
- Line mode output is not processed by the message automation table and is not written to the netlog. To obtain output from a command such as DISPSTAT in the netlog, use a PIPE command, for example:
  ```
  PIPE NETV DISPSTAT OUTMODE=LINE | LOGTO NETLOG
  ```
- Line mode output cannot be processed by a TRAP and WAIT.
- System operations commands can be issued within a NetView PIPE by using the OUTMODE=LINE parameter, unless noted otherwise in the command description.
- System operations commands supporting the OUTMODE=LINE option can be used in user-written clists. Note however, that the format of the output may change for follow-on Releases.
- If you work with OUTMODE=LINE no prompt panel is displayed.
- If no value is specified, the decision whether to display the command output by means of a full-screen panel or in line mode is based on the NetView task type the command is running on.

OUTDSN

This parameter lets you specify the name of the data set that is to contain the output of the command. You can specify a sequential data set or a member of a partitioned data set. The minimum record length is 80 bytes, except for the DISPSTAT and INGLIST commands where the minimum record length is 256 bytes. The data set must already exist. The OUTDSN parameter forces OUTMODE=LINE.

INGCF

Purpose

The INGCF command supports all the functions of msys for Operations that deal with coupling facilities. It supports full mode and line mode—for full line mode capability, refer to “INGCFL” on page 154. If you issue INGCF in line mode, only the display function is available. Therefore, you cannot start an action in msys for Operations when you issue INGCF from the NCCF console.

The INGCF command supports the following parameters:

- DRAIN

  Serves to remove all allocated structures from the coupling facility, to disconnect the coupling facility from the systems of the sysplex, and to inactivate the coupling facility.
- ENABLE

  Serves to activate a coupling facility, connect it with the systems of a sysplex and to populate it with structures.
- PATH

  Displays and controls the sender paths of the target coupling facility. It sets the sender paths ONLINE and OFFLINE physically and logically.
- STRUCTURE

  Displays detail information and rebuilds or deletes a selected structure on the target coupling facility. It also lets you start and stop duplexing.

INGCF associates a status with every coupling facility, and a condition with every structure (instance) that is allocated on the target coupling facility. The structure condition is influenced by the release level of the system that allocated the structure. The INGCF functions use the coupling facility state and the structure conditions to determine which action can be performed in any given situation. Therefore, the DRAIN and ENABLE functions can enforce a correct sequence of actions for complex tasks such as draining or restoring a coupling facility.

If the selected action impacts the sysplex configuration it must be confirmed before execution.

Authorizations

The actions that you can initiate with INGCF depend on your authorizations. The panels show your authorization type. Note that the authorization types apply to the current function, and that your authorization type may vary for different functions.

The following authorization types exist:

DISPLAY
You cannot initiate any action that affects the sysplex configuration.

ALL BUT (ACTIVATE|SHUTDOWN)
This type only occurs in the DRAIN and ENABLE command dialogs. You can rebuild structures, force the deletion of structures and set the sender paths offline and online, but you cannot inactivate or activate the coupling facility.

ALL
You can initiate all actions from the corresponding panel.

Depending on your authorizations, it is possible that you have, for example, authorization type ALL for the STRUCTURE function, and authorization type DISPLAY for the DRAIN function.

Syntax

```
►►──INGCF──┬─────────────────────────────────────────────────────────┬───────►
           └─DRAIN──┤ cfname ├───────────────────────────────────────┘
           ├─ENABLE──┤ cfname ├────────────────────────────────────┤
           ├─PATH──┤ cfname ├──────────────────────────────────────┤
           │                          ┌─CONDITION=NO──────────────┐ │
           └─STRUCTURE──┤ cfname ├──┼───────────────────────────┼─┘
                                      └─┬─CONDITION─┬──=──┬─NO──┬─┘
                                        └─COND──────┘     └─YES─┘

►──┬──────────────┬──────────────────────────────────────────────────────────►◄
   └─OUTMODE=LINE─┘
```

cfname:

```
├──┬─────────┬───────────────────────────────────────────────────────────┤
   └─CF_name─┘
```

Parameters

DRAIN
Prepares a coupling facility for removal from the sysplex.

ENABLE
(Re)integrates a coupling facility into a sysplex.

PATH
Controls the sender paths of a coupling facility.

STRUCTURE
Offers manipulation of individual structures (detail information, rebuild, deletion).

CF_name
Name of the target coupling facility for the specified function. The default is a selection panel that shows all available coupling facilities of the sysplex.

CONDITION
Specify YES if you want to get the current condition for each structure. Selecting this option increases the response time required to build the display. The default is NO.

OUTMODE
If you specify OUTMODE=LINE, INGCF is called in line mode. In this mode, only the display functions of the command are available.

Restrictions and Limitations

The ENABLE and the PATH functions require that the active IODF is catalogued. Otherwise, sender path information cannot be retrieved in certain situations.

INGCF ENABLE assumes that the receiver paths from the coupling facility to the systems of the sysplex have been defined and activated. This requires a POR of the CPC on which the coupling facility resides.

Coupling Facility States

The status of a coupling facility can be as follows:

ACTIVATING
The coupling facility is being activated and will then become DRAINED.

DEACTIVATING
The coupling facility is being deactivated and will then become INACTIVE.

DRAINING
The coupling facility is being disconnected from the connected systems.

DRAINED
The coupling facility has no connection to any system and can be removed from the sysplex.

DRAINED NOHWACC
The coupling facility has no connection to any system, but cannot be removed from the sysplex because the BCP (Basic Control Program) internal interface is not available.

Note: This status is also displayed when the coupling facility has been deactivated from the HMC (Hardware Management Console) but the XCF display commands still return the name of the coupling facility.

ENABLING
The coupling facility is being connected to the systems of the sysplex that use it.

FORCING
Allocated structures are being deleted from the coupling facility. This only happens with structures that have no active connectors, and with these only when they cannot be rebuilt by system-managed rebuild.

INACTIVE
The coupling facility is not active.

INACTIVE NOHWACC
The coupling facility is not active and cannot be activated because the BCP Internal Interface is not available.

NORMAL
The coupling facility may have allocated structures and is connected to all systems.

NORMAL OFFLINE
The coupling facility may have allocated structures. At least one system has set all its sender paths to this coupling facility to OFFLINE. XES will reject any rebuild request for this coupling facility.

NOTINPOLICY
The coupling facility is active but not defined in the active CFRM policy.

POPULATING
The coupling facility is being populated with all those structures that have it on the first place in their preference list.

REBUILDING
Either all allocated structures that can be rebuilt are being removed from the coupling facility by the XES rebuild process (initiated by DRAIN), or one particular such structure is being removed (initiated by the STRUCTURE).

Structure conditions

The condition of an allocated structure can be:

Rebuild is not supported.
The structure can neither be rebuilt, nor can its deletion be forced.

The structure has at least one active connector that does not support user-managed rebuild, and at least one active connector that does not support system-managed rebuild.

System-managed processes not supported.
The structure cannot be rebuilt, nor can its deletion be forced.

System-managed rebuild, which is a system-managed process, is not possible for one of the following reasons:

- The structure was allocated from a system with OS/390 2.7 or below.
- The CFRM couple data sets have not been formatted to support system-managed processes (`ITEM NAME(SMREBLD) NUMBER(1)` was not specified.).

Note: In certain rare cases system-managed processes are not supported although the condition that is displayed on the DRAIN panel seems to indicate the contrary. Then, the rebuild will be initiated, but will fail with message IXC367I indicating that system-managed processes are not supported for the structure.

No alternate coupling facility defined or available.
The structure can neither be rebuilt, nor can its deletion be forced.

The structure has an active connector and supports rebuild but has no alternate coupling facility defined in its preference list, or the alternate coupling facilities defined in the preference list are currently unavailable.

Insufficient space detected for rebuild.
The structure can or could not be rebuilt. Its deletion cannot be forced.

No alternate coupling facility has enough space to rebuild the structure.

Preference list is empty.
The structure cannot be rebuilt because its preference list is currently empty. A possible reason for this is a pending policy change; for pending policy changes, see “P column” on page 153.

Structure is pending deallocation.
XES accepted a forced deletion of the structure but does the real deallocation later.

Note: This status can only occur when MVS APAR OW39404 has not been installed.

Structure is being rebuilt.
The structure is being rebuilt to another coupling facility.

Duplex rebuild is being stopped.
Two instances of the structure were maintained on different coupling facilities. The application is being disconnected from that instance that is allocated on the target coupling facility. After disconnecting, the instance is deleted.

No connection exists.
The structure cannot be rebuilt, but you can force its deletion.

The structure has no connections and cannot be rebuilt with system-managed rebuild.

No alternate coupling facility for structure with no connections.
The structure cannot be rebuilt, but you can force its deletion.

The structure has no connections. It could be rebuilt with system-managed rebuild, but no alternate coupling facility is defined in its preference list or available.

No alternate coupling facility for structure with no active connections.
The structure cannot be rebuilt, but you can force its deletion.

The structure has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections. It could be rebuilt with system-managed rebuild, but no alternate coupling facility is defined in its preference list or available.

The structure's initial size is less than its actual size.
The structure can be rebuilt, but this can lead to loss of data.

An initial size is specified for the structure in the active CFRM policy. This initial size was used for the allocation of the structure. Afterwards, the size of the structure was increased either by the application itself or an operator command. However, the structure will only be rebuilt with its initial size. Subsequently, INGCF will expand the structure to its actual size again, but this will happen *after* the data have been transferred. To avoid a potential loss of data, the application has to change the initial size to the actual size.

No active connection exists.
The structure cannot be rebuilt, but you can force its deletion.

The structure has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections and cannot be rebuilt with system-managed rebuild.

Note: INGCF DRAIN deallocates structures with this condition as part of the REBUILD action (see "REBUILD (F10)" on page 141). INGCF STRUCTURE accepts a rebuild request for structures with this condition, but deallocates them (see "Rebuild (R)" on page 152).

No connections. System-managed rebuild supported.
The structure can be rebuilt.

The structure has no connections, but can be rebuilt with system-managed rebuild.

No active connections. System-managed rebuild supported.
The structure can be rebuilt with system-managed rebuild.

User-managed rebuild is not possible for the structure because it has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections.

System-managed rebuild is supported
The structure can be rebuilt.

The structure has active connectors. At least one active connector does not support user-managed rebuild, but all active connectors support system-managed rebuild.

Duplex rebuild is active.
The application is connected to two instances of the same structure on different coupling facilities.

[No condition]
When no condition is displayed, the structure can be rebuilt.

The structure has at least one active connection, and all its active connectors support user-managed rebuild.

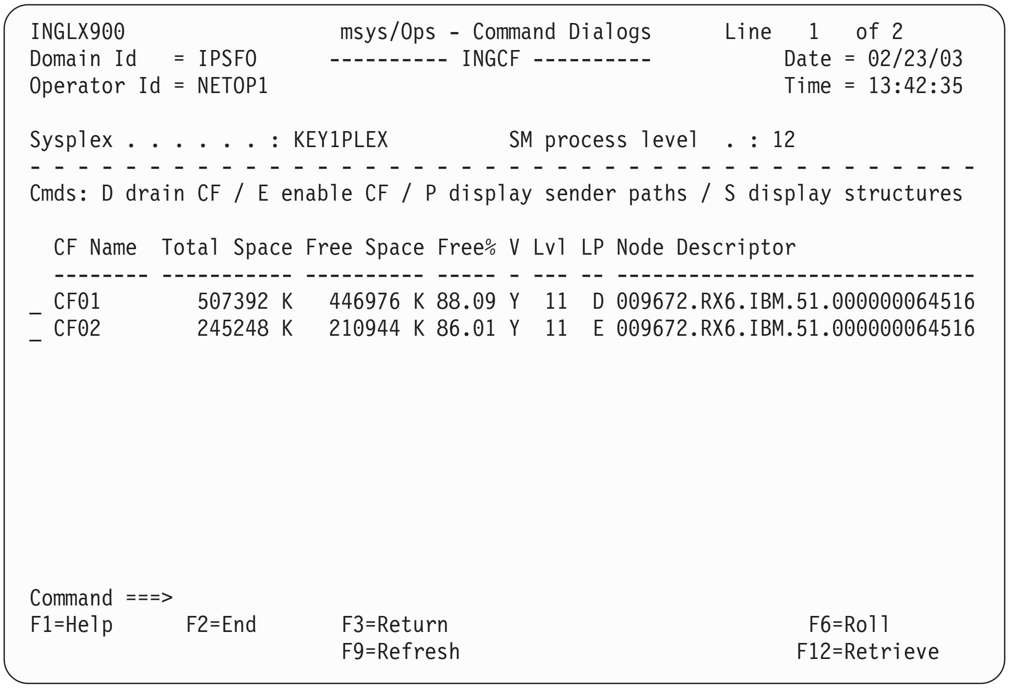
Structure is awaiting rebuild.
The structure has been selected for rebuild but has not been processed yet.

Structure is currently allocated on ***cf_name*****.**
The structure can be rebuilt on the target coupling facility with the POPULATE action of the ENABLE function. It is currently allocated on the *cf_name* coupling facility, but the target coupling facility precedes *cf_name* in the structure's preference list. This condition is displayed only in the ENABLE command dialog.

Structure allocated in *cf_name* **cannot be rebuilt to this CF.**
The structure can probably not be rebuilt on the target CF with the POPULATE action of the ENABLE function. It is currently allocated in the *cf_name* CF, but the target CF precedes *cf_name* in the structure's preference list. And, the actual size of the structure is greater than the free space of the target CF. This condition is displayed only in the ENABLE command dialog.

Example

If you issue INGCF without any parameters, a panel with all coupling facilities of the sysplex is displayed:



```
INGLX900                 msys/Ops - Command Dialogs      Line   1   of 2
Domain Id   = IPSFO      ---------- INGCF ----------          Date = 02/23/03
Operator Id = NETOP1                                          Time = 13:42:35

Sysplex . . . . . . : KEY1PLEX         SM process level  . : 12
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: D drain CF / E enable CF / P display sender paths / S display structures

  CF Name  Total Space Free Space Free% V Lvl LP Node Descriptor
  -------- ----------- ---------- ----- - --- -- ------------------------------
_ CF01        507392 K   446976 K 88.09 Y  11  D 009672.RX6.IBM.51.000000064516
_ CF02        245248 K   210944 K 86.01 Y  11  E 009672.RX6.IBM.51.000000064516









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

Figure 11. INGCF selection panel

Specify a function for a selected coupling facility and press Enter.

DRAIN

Purpose

The DRAIN function of INGCF facilitates the removal of a coupling facility from the sysplex, for example, for maintenance purposes. With this option, you can perform the following sequence of tasks:

1. Display information for all allocated structures of the coupling facility.
2. Rebuild all rebuildable structures on *another* coupling facility, and delete instances of structures on the target coupling facility that are being duplexed on another coupling facility.

   Notes:

   a. The scope of the structures that can be rebuilt depends on the release level of the sysplex members.
   b. INGCF DRAIN rebuilds structures one at a time (SETXCF START,REBUILD,STRNAME=), not globally (SETXCF START,REBUILD,CFNAME=), and always on a coupling facility that is different from the target coupling facility (LOCATION=OTHER).
   c. Generally, you should be aware that it is XES that performs the actual rebuild. Not all of the factors that XES takes into account when allocating a structure are accessible to msys for Operations. Therefore, a rebuild request for a structure that should be rebuildable according to its condition can fail in certain rare cases.
3. Force the deletion of structures that have no active connectors and cannot be rebuilt.

Note that there are structures that you can neither rebuild nor delete with the force action. These include the structures that have at least one active connector and do not support rebuild. To remove such structures first disconnect all active connectors, and then delete the structure manually if it is persistent or has persistent connections.

4. Disconnect the coupling facility from the systems with which it is connected.
5. Inactivate the target coupling facility.

INGCF DRAIN ensures that these actions are performed in the correct order, as specified above.

Actions

The following F-keys are supported:

***REBUILD (F10)**

Starts the rebuild of structures that can be rebuilt on *another* coupling facility. Thus, a rebuild is only initiated for structures whose preference list contains more than one coupling facility.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems that have OS/390 2.8 or above; it must have been enabled by formatting the CFRM couple data sets with the specification

```
ITEM NAME(SMREBLD) NUMBER(1)
```

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

The REBUILD action also deletes all structure instances on the target coupling facility that are being duplexed on another coupling facility.

Note: The REBUILD action *deallocates* structures with the condition 'No active connection exists.'. See “No active connection exists” on page 139.

***FORCE (F5)**
Forces the deallocation of structures with one of the following conditions:

- `No connection exists.`
- `No alternate coupling facility for structure with no active connections.`
- `No alternate coupling facility for structure with no connections.`

This action is only made available after all structures that can be rebuilt have been rebuilt.

***DRAIN (F4)**
Disconnects the coupling facility from its connected systems by setting the sender path(s) OFFLINE.

This action is only enabled after all structures of the target coupling facility have been removed to another coupling facility or deallocated. Note that structures that have active connectors but do not support rebuild cannot be removed with F10 or F5. They must be deallocated manually before executing this step is enabled.

***SHUTDOWN (F11)**
This action inactivates the coupling facility. It is only made available when all connections between the coupling facility and the systems of the sysplex have been disconnected.

Note: This function key is unavailable when running on a z/OS image which runs under z/VM.

Note that these actions can only be performed if INGCF DRAIN is issued in full mode. In line mode, only the display function is available.

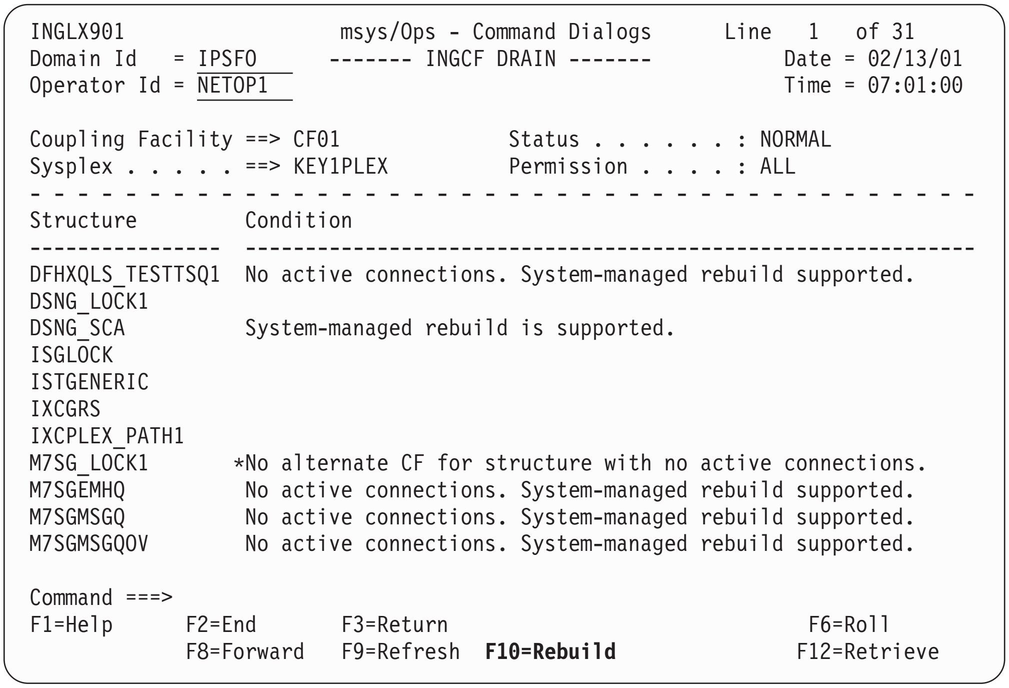
To avoid performance degradation due to multiple rebuild processes, or unpredictable results due to multiple executions of an action, all actions are locked. Therefore, an action is rejected if any lock exists even if the action does not affect the action currently being performed. Because the action can take a long time it is also executed asynchronously on a dedicated autotask, preventing the operator from being blocked. To check progress, use the refresh function (F9).

Example

In the following example, a coupling facility is drained:

1. All of its structures that can be rebuilt are rebuilt on another coupling facility, and duplexing is stopped.
2. For all structures that have no active connector and cannot be rebuilt deletion is forced.
3. All systems that are connected with the coupling facility are disconnected.
4. The coupling facility is inactivated.

When you issue INGCF with the option DRAIN, you can specify the coupling facility to be drained, for example by entering `INGCF DRAIN CF01`; in this case, the panel of Figure 12 on page 143 is displayed at once. If you do not specify a coupling facility name, INGCF displays a selection panel with all coupling facilities that are defined in the sysplex. After selection of CF01, INGCF displays the following panel:

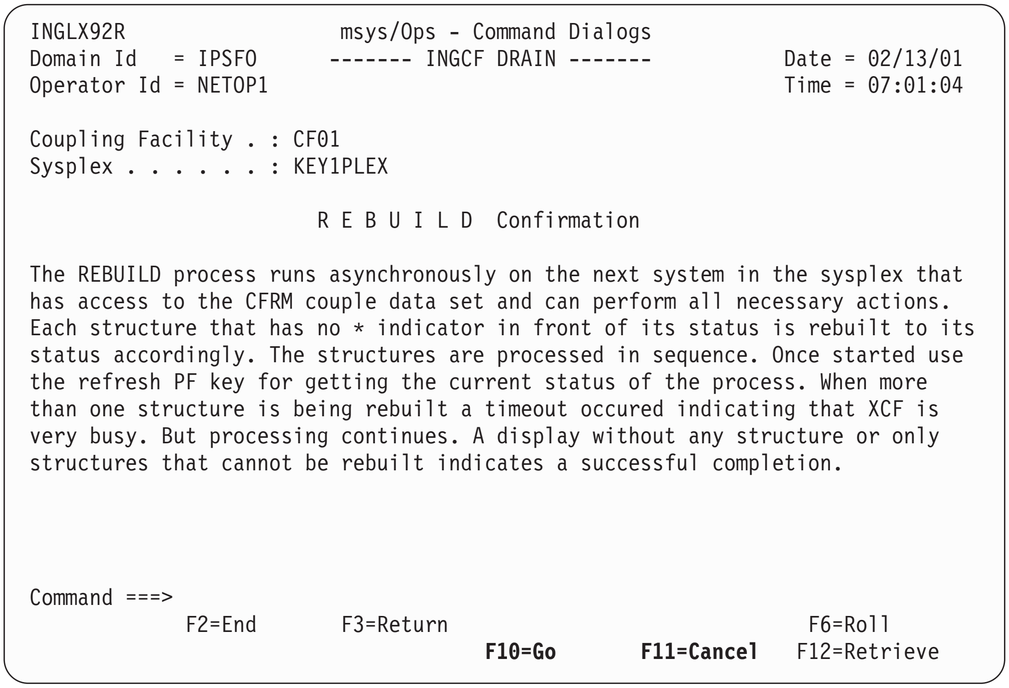


```
INGLX901                  msys/Ops - Command Dialogs      Line   1   of 31
Domain Id   = IPSFO      ------- INGCF DRAIN -------          Date = 02/13/01
Operator Id = NETOP1                                          Time = 07:01:00

Coupling Facility ==> CF01              Status . . . . . . : NORMAL
Sysplex . . . . . ==> KEY1PLEX          Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure         Condition
----------------  -----------------------------------------------------------
DFHXQLS_TESTTSQ1  No active connections. System-managed rebuild supported.
DSNG_LOCK1
DSNG_SCA          System-managed rebuild is supported.
ISGLOCK
ISTGENERIC
IXCGRS
IXCPLEX_PATH1
M7SG_LOCK1       *No alternate CF for structure with no active connections.
M7SGEMHQ          No active connections. System-managed rebuild supported.
M7SGMSGQ          No active connections. System-managed rebuild supported.
M7SGMSGQOV        No active connections. System-managed rebuild supported.

Command ===>
F1=Help      F2=End       F3=Return                              F6=Roll
             F8=Forward   F9=Refresh  F10=Rebuild               F12=Retrieve
```

Figure 12. DRAIN command dialog panel: before any action

The status of the coupling facility (NORMAL) and the authorization type of the operator (ALL) are displayed on the right side of the panel header. The main part of the panel consists of a list containing the structures allocated in CF01 and their conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

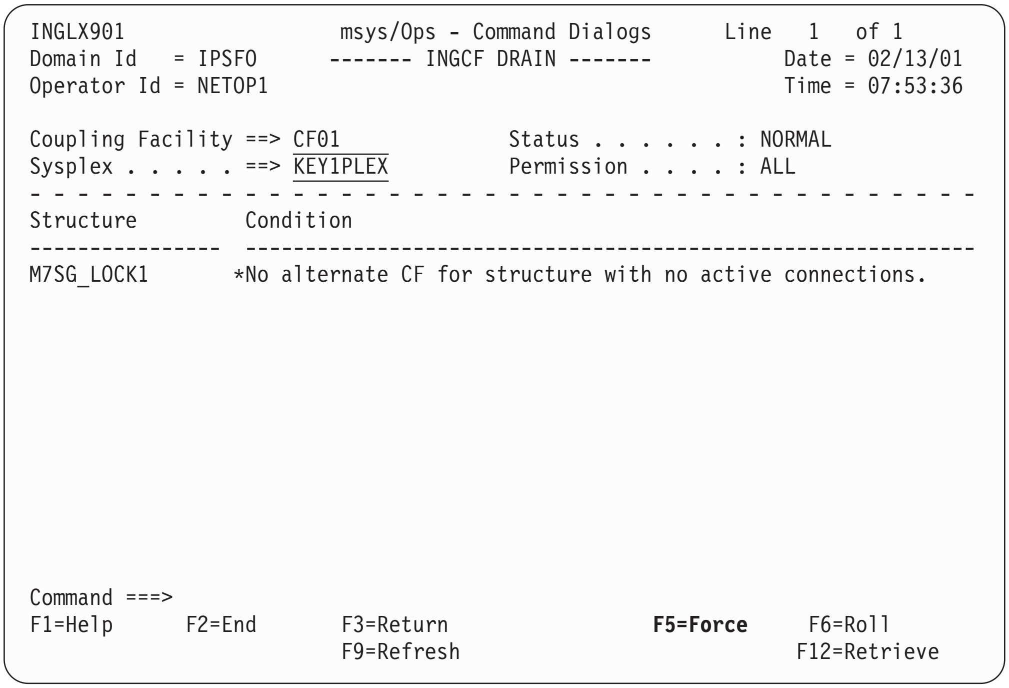
The only action that is enabled is REBUILD with F10. Pressing F10 calls the following confirmation panel:



```
INGLX92R                 msys/Ops - Command Dialogs
Domain Id   = IPSFO      ------- INGCF DRAIN -------          Date = 02/13/01
Operator Id = NETOP1                                          Time = 07:01:04

Coupling Facility . : CF01
Sysplex . . . . . . : KEY1PLEX

                        R E B U I L D  Confirmation

The REBUILD process runs asynchronously on the next system in the sysplex that
has access to the CFRM couple data set and can perform all necessary actions.
Each structure that has no * indicator in front of its status is rebuilt to its
status accordingly. The structures are processed in sequence. Once started use
the refresh PF key for getting the current status of the process. When more
than one structure is being rebuilt a timeout occured indicating that XCF is
very busy. But processing continues. A display without any structure or only
structures that cannot be rebuilt indicates a successful completion.




Command ===>
             F2=End       F3=Return                            F6=Roll
                                    F10=Go       F11=Cancel   F12=Retrieve
```

Figure 13. DRAIN command dialog: confirmation panel for REBUILD

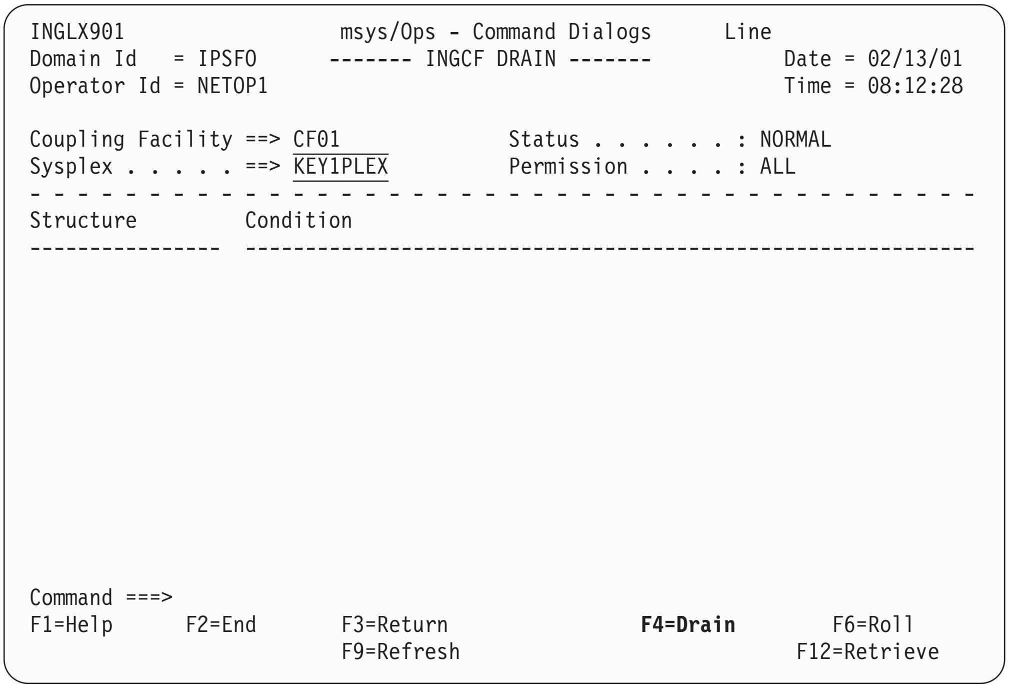
After F10 was pressed and the rebuild is complete the command dialog can be refreshed with F9. It looks as follows:



```
INGLX901                 msys/Ops - Command Dialogs      Line   1   of 1
Domain Id   = IPSFO      ------- INGCF DRAIN -------          Date = 02/13/01
Operator Id = NETOP1                                          Time = 07:53:36

Coupling Facility ==> CF01             Status . . . . . . : NORMAL
Sysplex . . . . . ==> KEY1PLEX         Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure         Condition
----------------  -----------------------------------------------------------
M7SG_LOCK1       *No alternate CF for structure with no active connections.










Command ===>
F1=Help      F2=End       F3=Return               F5=Force     F6=Roll
                          F9=Refresh                          F12=Retrieve
```

Figure 14. DRAIN command dialog panel: after rebuild

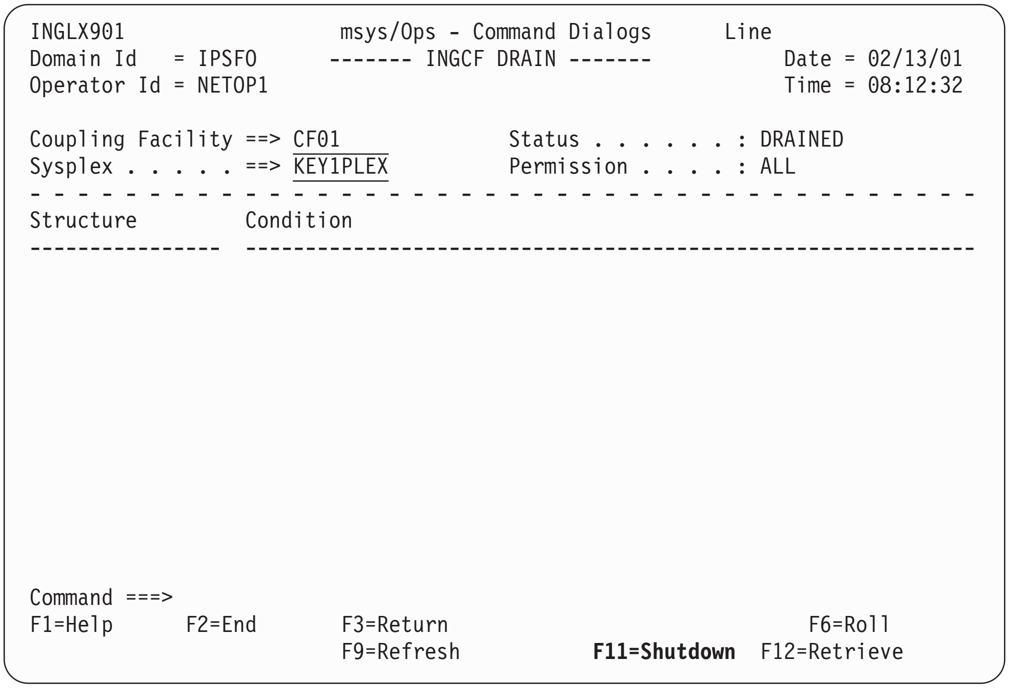
One structure could not be rebuilt because no alternate coupling facility is specified in its preference list. The REBUILD action is no longer available. Instead, the FORCE action (F5) is available because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure. Pressing F5 calls a confirmation panel similar to that for REBUILD. Pressing F10 on the confirmation panel and refreshing the command dialog after the action has been completed results in the following panel:



```
INGLX901                 msys/Ops - Command Dialogs      Line
Domain Id   = IPSFO     ------- INGCF DRAIN -------          Date = 02/13/01
Operator Id = NETOP1                                         Time = 08:12:28

Coupling Facility ==> CF01            Status . . . . . . : NORMAL
Sysplex . . . . . ==> KEY1PLEX        Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure        Condition
---------------- ---------------------------------------------------------------









Command ===>
F1=Help      F2=End       F3=Return               F4=Drain        F6=Roll
                          F9=Refresh                            F12=Retrieve
```

Figure 15. DRAIN command dialog panel: after forcing

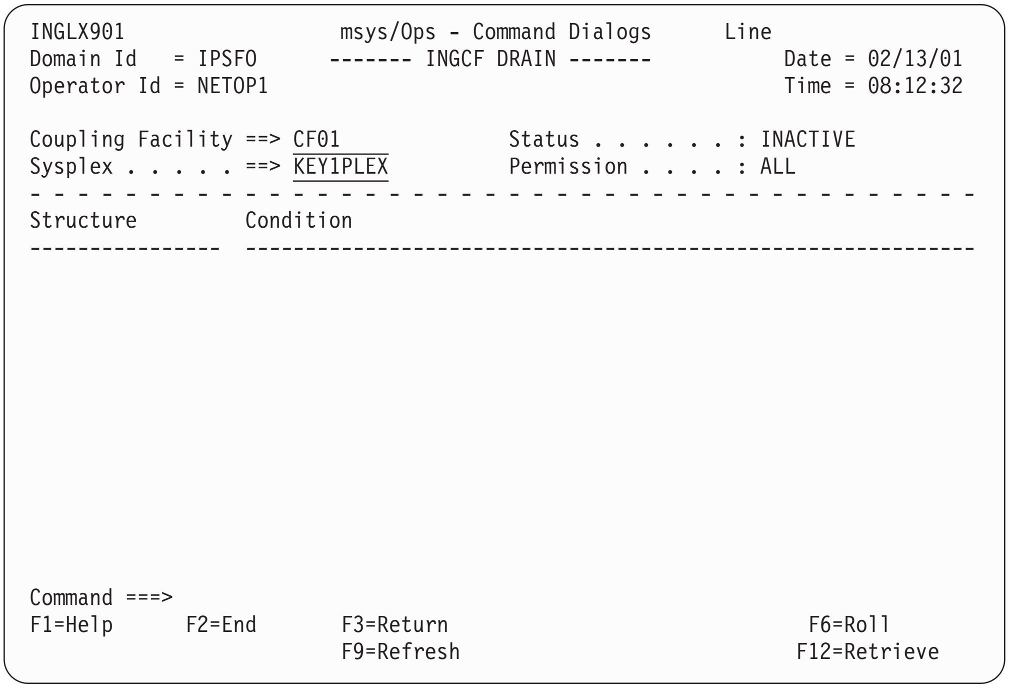
No more structures are allocated in the coupling facility, so that the coupling facility can be released from the connections with the systems of the sysplex. Consequently, INGCF DRAIN enables the DRAIN action (F4). After completion of that action, the status of the coupling facility changes to DRAINED, as shown on the following panel:



```
INGLX901                  msys/Ops - Command Dialogs      Line
Domain Id   = IPSFO      ------- INGCF DRAIN -------          Date = 02/13/01
Operator Id = NETOP1                                          Time = 08:12:32

Coupling Facility ==> CF01             Status . . . . . . : DRAINED
Sysplex . . . . . ==> KEY1PLEX         Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure        Condition
----------------  -------------------------------------------------------------














Command ===>
F1=Help      F2=End       F3=Return                            F6=Roll
                          F9=Refresh          F11=Shutdown  F12=Retrieve
```

Figure 16. DRAIN command dialog panel: panel after draining

Because the coupling facility is no longer connected to any system, it can be inactivated. After pressing F11 the status of the coupling facility changes to INACTIVE:



```
INGLX901                  msys/Ops - Command Dialogs      Line
Domain Id   = IPSFO      ------- INGCF DRAIN -------          Date = 02/13/01
Operator Id = NETOP1                                          Time = 08:12:32

Coupling Facility ==> CF01             Status . . . . . . : INACTIVE
Sysplex . . . . . ==> KEY1PLEX         Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure        Condition
----------------  -------------------------------------------------------------














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

Figure 17. DRAIN command dialog panel: after inactivation

ENABLE

Purpose

The ENABLE function of the INGCF command is intended to support the integration AND re-integration of a coupling facility into a sysplex. With this option, you can:

1. Activate the target coupling facility.
2. Connect the systems of the sysplex with the coupling facility.
3. Switch to another CFRM policy if the target coupling facility is not defined in the active policy and a suitable policy is available.

   A suitable CFRM policy must contain:

   - A definition of the target coupling facility
   - Appropriate definitions for every active coupling facility and every allocated structure
4. Rebuild all structures on the target coupling facility whose preference list starts with this coupling facility, unless this is excluded by other requirements.

INGCF ENABLE ensures that these actions are performed in the correct order, as specified above.

Actions

The possible actions and the associated F-keys are:

***ACTIVATE (F11)**
This action activates the CFCC (Coupling Facility Control Code) through the BCP Internal Interface by an ACTIVATE command.

Note: This function key is unavailable when running on a z/OS image which runs under z/VM.

***ENABLE (F4)**
Sets the sender path(s) of all systems of the sysplex to ONLINE. This action is enabled when the coupling facility is active.

***SWITCH (F5)**
Switches to another CFRM policy when the target coupling facility is not defined in the active CFRM policy and a suitable policy is available. When there is more than one suitable policy you can choose one of these from a selection panel.

A CFRM policy is suitable when it contains:

- A definition of the target coupling facility
- Definitions for every active coupling facility and every allocated structure

This action is only made available when the target coupling facility is active, but not defined in the current CFRM policy.

***POPULATE (F10)**
Starts a rebuild process by which all structures that have the target coupling facility at the beginning of their preference list but are currently allocated on another coupling facility are allocated on the target coupling facility.

This action requires that the coupling facility be enabled, connected to all members of the sysplex, and defined in the current CFRM policy. The action is offered whenever INGCF ENABLE detects that a structure is not allocated on the target coupling facility although it is the preferred coupling facility of that structure.

Note: When you have drained a coupling facility with INGCF DRAIN and then reintegrate it with INGCF ENABLE, be aware that the set of structures that are allocated on the target coupling facility after population can be different from the original set before the draining. Typically, this happens when the original set does not contain exactly those structures that have the target coupling facility at the first position in their preference list.

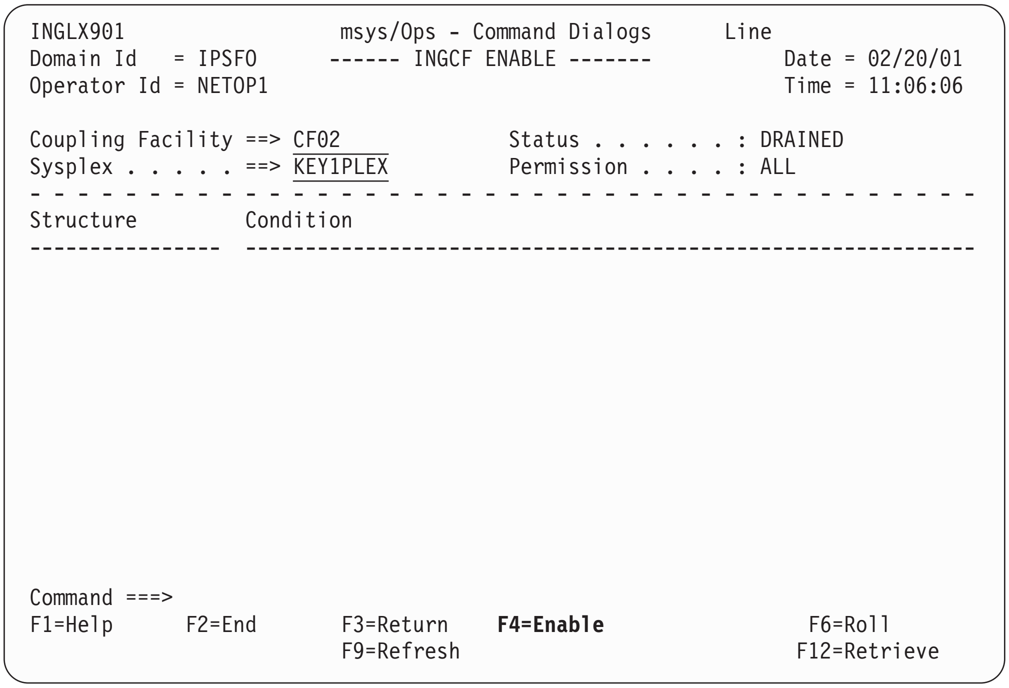
Note that these actions can only be performed when INGCF ENABLE is called in full mode. In line mode, only the display function is available.

Example

In the following example, a coupling facility that has already been activated is reintegrated into the sysplex in two steps:

1. The coupling facility is connected to all systems of the sysplex.
2. All structures that have the target coupling facility as the first coupling facility in their preference list are allocated on the target coupling facility.

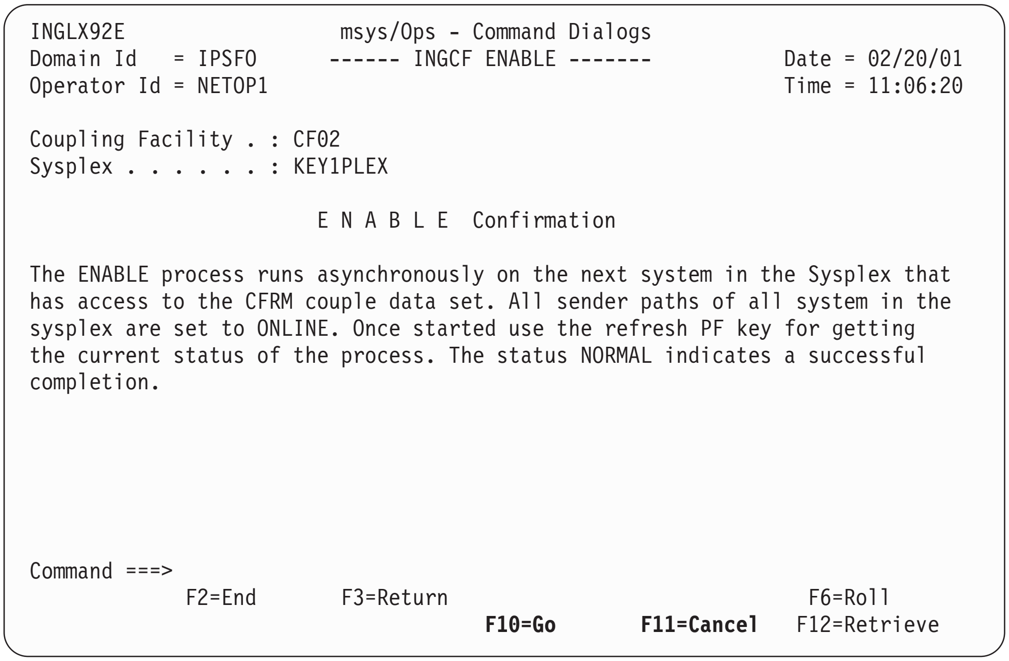
If you issue INGCF with the option ENABLE, you can specify the coupling facility to be reintegrated, for example by entering `INGCF ENABLE CF02`. In this case, the panel of Figure 18 is displayed at once. If you do not specify a coupling facility name, INGCF shows a selection panel with all coupling facilities that are defined in the sysplex. After selection of CF02, INGCF displays the following panel:



```
INGLX901                  msys/Ops - Command Dialogs      Line
Domain Id   = IPSFO      ------ INGCF ENABLE -------          Date = 02/20/01
Operator Id = NETOP1                                          Time = 11:06:06

Coupling Facility ==> CF02            Status . . . . . . : DRAINED
Sysplex . . . . . ==> KEY1PLEX        Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure        Condition
---------------  -------------------------------------------------------------












Command ===>
F1=Help      F2=End       F3=Return    F4=Enable                 F6=Roll
                          F9=Refresh                            F12=Retrieve
```

Figure 18. ENABLE command dialog panel: before any action

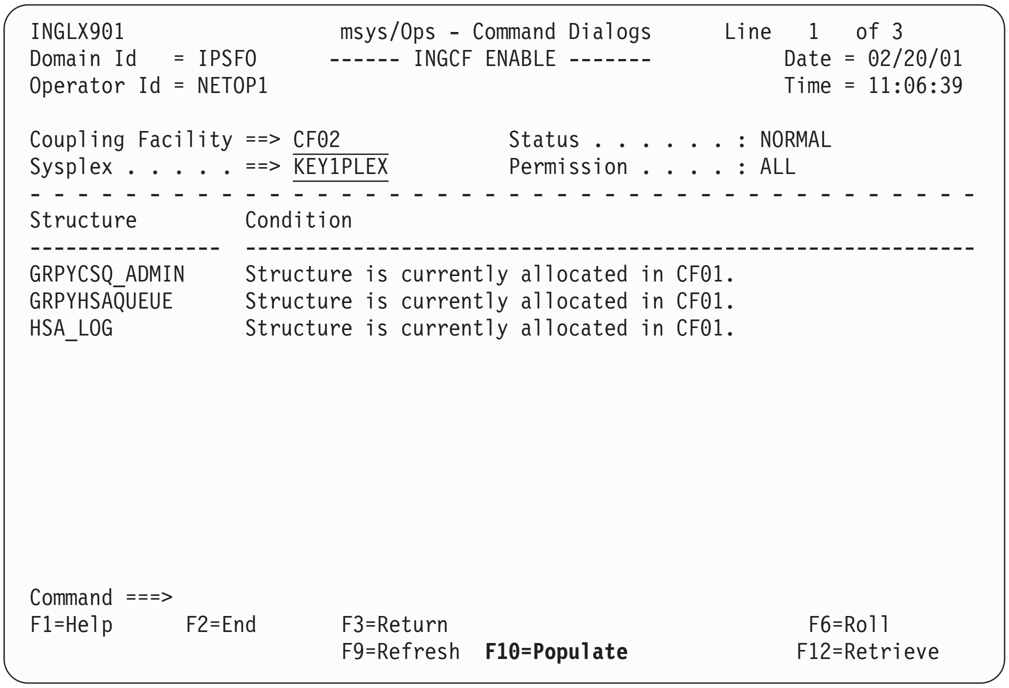
The selected coupling facility has already been activated manually, therefore its status, as shown on the right side of the panel header, is `DRAINED`. The authorization type of the operator (`ALL`) is also displayed on the right side of the panel header. The main part of the panel is empty because no structures are allocated in CF02. The only action that is activated is ENABLE with F4. If you press F4 the following confirmation panel is displayed:



```
INGLX92E                  msys/Ops - Command Dialogs
Domain Id   = IPSFO      ------ INGCF ENABLE -------           Date = 02/20/01
Operator Id = NETOP1                                           Time = 11:06:20

Coupling Facility . : CF02
Sysplex . . . . . . : KEY1PLEX

                        E N A B L E  Confirmation

The ENABLE process runs asynchronously on the next system in the Sysplex that
has access to the CFRM couple data set. All sender paths of all system in the
sysplex are set to ONLINE. Once started use the refresh PF key for getting
the current status of the process. The status NORMAL indicates a successful
completion.






Command ===>
             F2=End       F3=Return                               F6=Roll
                                      F10=Go       F11=Cancel   F12=Retrieve
```

Figure 19. Confirmation panel for ENABLE

After pressing F10 on the confirmation panel, the command dialog changes as follows:



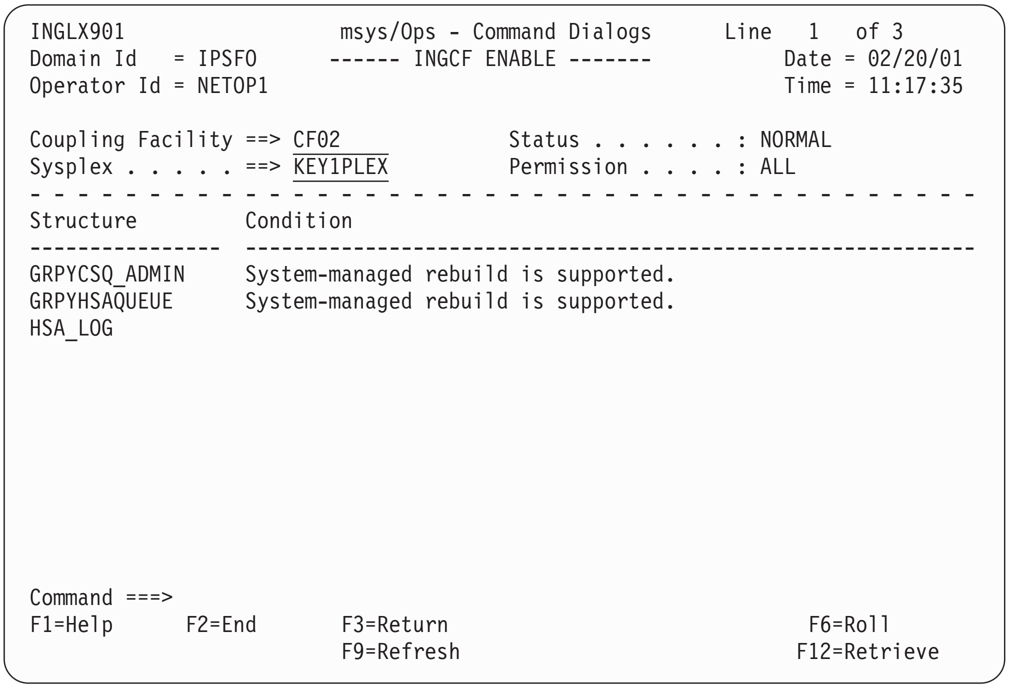
```
INGLX901                  msys/Ops - Command Dialogs      Line   1   of 3
Domain Id   = IPSFO      ------ INGCF ENABLE -------           Date = 02/20/01
Operator Id = NETOP1                                           Time = 11:06:39

Coupling Facility ==> CF02              Status . . . . . . : NORMAL
Sysplex . . . . . ==> KEY1PLEX          Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure         Condition
----------------  --------------------------------------------------------------
GRPYCSQ_ADMIN     Structure is currently allocated in CF01.
GRPYHSAQUEUE      Structure is currently allocated in CF01.
HSA_LOG           Structure is currently allocated in CF01.









Command ===>
F1=Help      F2=End       F3=Return                               F6=Roll
                          F9=Refresh  F10=Populate              F12=Retrieve
```

Figure 20. ENABLE command dialog panel: after enabling

The status has changed to NORMAL, and F10 is enabled for populating the coupling facility. This implies that the target coupling facility is defined in the active CFRM policy.

The structure list contains three entries with the condition 'Structure is currently allocated in CF01.'. These are the structures that are currently allocated in CF01, but have CF02 at the first position in their preference list.

Pressing F10 populates the coupling facility, and the refreshed panel looks as follows:



```
INGLX901                 msys/Ops - Command Dialogs      Line   1   of 3
Domain Id   = IPSFO      ------ INGCF ENABLE -------          Date = 02/20/01
Operator Id = NETOP1                                          Time = 11:17:35

Coupling Facility ==> CF02            Status . . . . . . : NORMAL
Sysplex . . . . . ==> KEY1PLEX        Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure         Condition
----------------  -----------------------------------------------------------
GRPYCSQ_ADMIN     System-managed rebuild is supported.
GRPYHSAQUEUE      System-managed rebuild is supported.
HSA_LOG











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

Figure 21. ENABLE command dialog panel: after populating

The POPULATE action is no longer available because all structures whose preference list starts with CF02 are allocated in CF02.

PATH

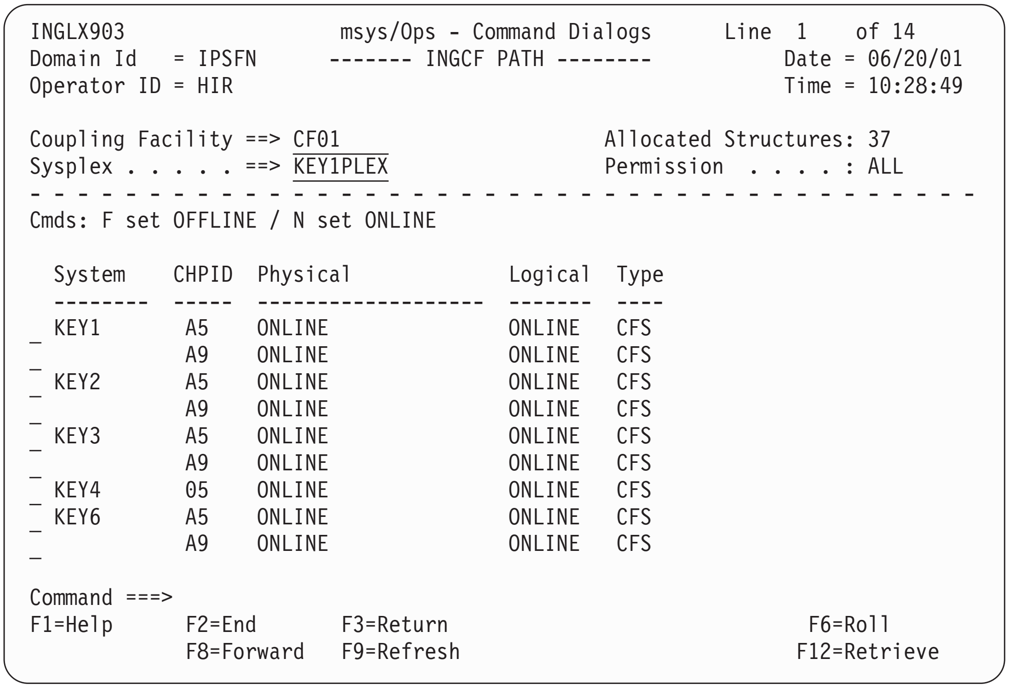
Purpose

The INGCF PATH function displays the sender paths, that is, the paths from the connected systems to the specified coupling facility.

Restrictions

The last sender path of each system can only be set to OFFLINE when no more structures are allocated.

Example



```
INGLX903                  msys/Ops - Command Dialogs      Line  1    of 14
Domain Id   = IPSFN      ------- INGCF PATH --------          Date = 06/20/01
Operator ID = HIR                                             Time = 10:28:49

Coupling Facility ==> CF01                        Allocated Structures: 37
Sysplex . . . . . ==> KEY1PLEX                    Permission  . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: F set OFFLINE / N set ONLINE

  System    CHPID  Physical             Logical  Type
  --------  -----  -------------------  -------  ----
_ KEY1       A5    ONLINE               ONLINE   CFS
_            A9    ONLINE               ONLINE   CFS
_ KEY2       A5    ONLINE               ONLINE   CFS
_            A9    ONLINE               ONLINE   CFS
_ KEY3       A5    ONLINE               ONLINE   CFS
_            A9    ONLINE               ONLINE   CFS
_ KEY4       05    ONLINE               ONLINE   CFS
_ KEY6       A5    ONLINE               ONLINE   CFS
_            A9    ONLINE               ONLINE   CFS

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

Figure 22. PATH command dialog panel

The following command codes are available:

F Sets the sender path OFFLINE.

N Sets the sender path ONLINE.

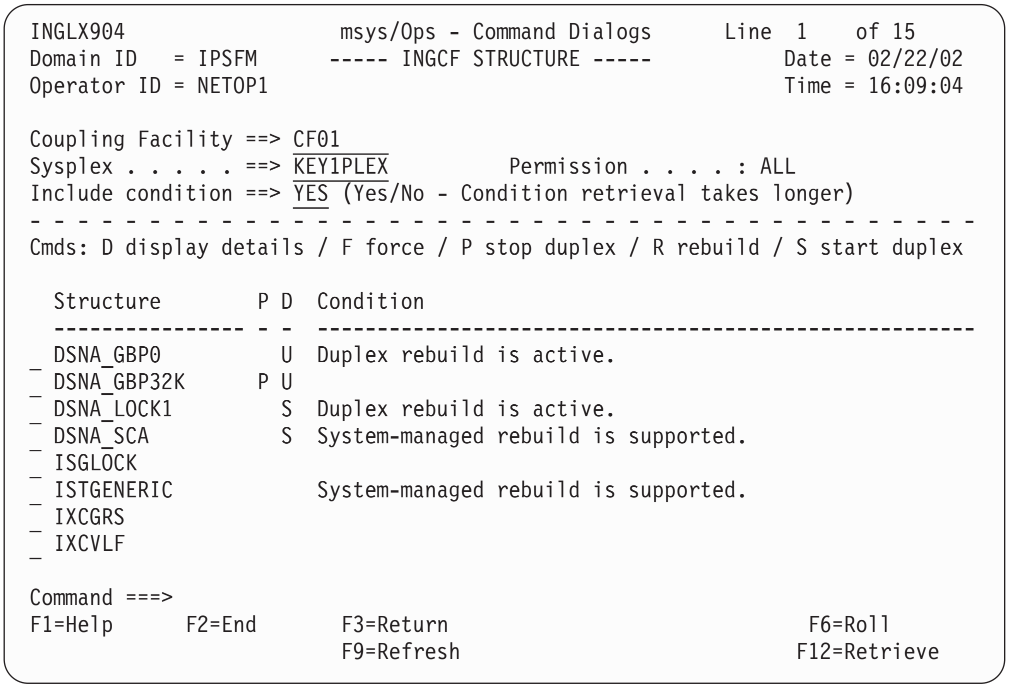
- If you have issued INGCF with the PATH parameter, the **Coupling Facility** field is an input field. To display the path list of another coupling facility specify the name of the coupling facility in this field and press ENTER.
- The **Allocated Structures** field shows the number of allocated structures.
- The **Permission** field shows your authorization level.
- The **System** column contains the names of the systems that are connected to the target coupling facility.
- The **CHPID** column shows the IDs of the sender channel paths.
- The **Physical** column shows the status of the sender channel paths.
- The **Logical** column shows the logical status of the paths to that coupling facility.
- The **Type** column shows the type of the sender channel paths.

STRUCTURE

Purpose

The STRUCTURE function of the INGCF displays the allocated structures of a coupling facility. You can initiate a rebuild or deallocation of a selected structure if the conditions for these actions are satisfied.

Example



```
INGLX904                  msys/Ops - Command Dialogs      Line  1    of 15
Domain ID   = IPSFM      ----- INGCF STRUCTURE -----           Date = 02/22/02
Operator ID = NETOP1                                           Time = 16:09:04

Coupling Facility ==> CF01
Sysplex . . . . . ==> KEY1PLEX          Permission . . . . : ALL
Include condition ==> YES (Yes/No - Condition retrieval takes longer)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: D display details / F force / P stop duplex / R rebuild / S start duplex

  Structure        P D  Condition
  ---------------- - -  -------------------------------------------------------
_ DSNA_GBP0          U  Duplex rebuild is active.
_ DSNA_GBP32K      P U
_ DSNA_LOCK1         S  Duplex rebuild is active.
_ DSNA_SCA           S  System-managed rebuild is supported.
_ ISGLOCK
_ ISTGENERIC            System-managed rebuild is supported.
_ IXCGRS
_ IXCVLF
_

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

Figure 23. STRUCTURE command dialog panel

The following command codes are available:

D Displays detail information about the structure.

***F** Forces the deallocation of the structure if it has one of the following conditions:

- `No connection exists.`
- `No alternate CF for structure with no active connections.`
- `No alternate CF for structure with no connections.`

When you try to force the deallocation of a structure that can be rebuilt, an error message is issued.

***P** Stops duplexing of the selected structure.

***R** Starts the rebuild of the selected structure. Depending on the PENDING status, the automation starts the rebuild with a different LOCATION parameter (PENDING uses the parameter LOCATION=NORMAL, otherwise LOCATION=OTHER). A rebuild with the parameter LOCATION=OTHER is only initiated for structures whose preference list contains more than one coupling facility.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems that have OS/390 2.8 and above ; it must have been enabled by formatting the CFRM couple data sets with the specification

```
ITEM NAME(SMREBLD) NUMBER(1)
```

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

INGCF STRUCTURE accepts a rebuild request for structures with the condition 'No active connection exists.', but *deallocates* them. See "No active connection exists" on page 139.

***S** Starts duplexing of the selected structure.

There are two methods for duplexing, user-managed and system-managed duplexing. User-managed duplexing is supported for all release levels. System-managed duplexing is only available when all systems in the Parallel Sysplex have been upgraded to z/OS 1.2 or later with APAR OW41617, and appropriate APARs listed in the CFDUPLEX PSP bucket (for more information, see *System-Managed CF Structure Duplexing*, GM13-0103-03). System-managed duplexing must have been enabled by formatting the CFRM couple data sets with the specification

```
ITEM NAME(SMDUPLEX) NUMBER(1)
```

System-managed duplexing is only performed when the requirements for user-managed duplexing are not met. This applies, for example, to structures without active connectors.

Starting the duplex rebuild of a structure requires at least the policy entry allowing the duplex rebuild of the structure. If there is no entry the duplex rebuild is disabled. The other requirements depend on the type of the duplex rebuild. When all connectors to a structure allow user-managed duplex rebuild, this type takes precedence over system-managed duplex rebuild. However, user-managed rebuild also requires at least one active connector. Thus, when the operator starts the duplex rebuild for a structure allowing user-managed duplex rebuild as well as system-managed rebuild but without having active connectors, XCF tries to initiate a system-managed duplex rebuild. System-managed duplex rebuild has the following requirements:

- System-managed rebuild must be supported by all connectors.
- The structure must be allocated in a coupling facility supporting system-managed duplexing and another coupling facility supporting system-managed duplexing must be defined in its preference list.
- The CFRM couple data set must support system-managed duplex rebuild and the structure must not have a policy change pending.
- The structure must be defined in the active CFRM policy when any connection state is not active.

- If you have specified INGCF with the STR parameter, the **Coupling Facility** field is an input field. To display the structure list of another coupling facility, specify the name of the coupling facility in this field and press Enter.
- The **Include Condition** field is an input field. By specifying Yes or No in this field you determine whether or not the conditions of the structures are displayed in the **Structure** column.
- The **Permission** field shows your authorization level. There are two possible values, ALL and DISPLAY. DISPLAY signifies that you can only use the display functions. ALL signifies that you can also rebuild and delete structures.
- You can specify an action code before every structure entry. The codes you can enter depend on your authorization level
- The **Structure** column shows the names of the structures.
- The letter P in the **P** column indicates that policy changes are pending for the structure.

A structure has policy changes pending when it was allocated at the time of a CFRM policy switch, and XES could not bring the switch into effect for that structure. One reason for a pending policy change is that the old and the new policy define the structure differently, for example, with different preference lists.

- The **Condition** column shows the status of the structures. You can switch the display of the conditions on and off with the **Include Condition** field.
- The D field indicates the type of duplexing that is possible. The following values are possible:

  U User-managed duplexing

  S System-managed duplexing

  B User-managed and system-managed duplexing

INGCFL

Purpose

The INGCFL routine supports line mode for INGCF other than display capability. For further information refer to "INGCF" on page 134.

Syntax

```
                                     ┌─RESP──=──SYNC──┐
►►──INGCFL──┬─ACTIVATE───┬──CF_name──┼────────────────┼──────────────────►◄
            ├─DEACTIVATE─┤           └─RESP──=──ASYNC─┘
            ├─DRAIN──────┤
            ├─ENABLE─────┤
            ├─POPULATE───┤
            └─REBUILD────┘
```

Parameters

cfname
Is the name of the CF

ACTIVATE
Activates the coupling facility

DEACTIVATE
Deactivates the coupling facility after performing DRAIN

DRAIN
Sets the sender paths to OFFLINE after performing a REBUILD

ENABLE
Sets the sender paths to ONLINE after performing ACTIVATE

POPULATE
Starts the populate process of the coupling facility after performing ENABLE

REBUILD
Starts the rebuild process of the coupling facility

RESP
Specifies whether the final result is returned synchronously via return code or asynchronously via message (default: synchronous response)

Note: The real activation and deactivation of a coupling facility are unavailable when running on a z/OS image which runs under z/VM.

INGHC

Purpose

This command serves two purposes:

1. Allows you to view the reports of the IBM Health Checker for z/OS and Sysplex (HealthChecker)
2. Allows you to request the following actions from the HealthChecker:
   - Perform individual checks according to your filtering options
   - Override IBM's best practices with your own values

Types of reports

About the reports

The following rule applies for each check overridden in the user policy: If there is some error in your specification, the check is not performed at all, that is, the IBM values are not used either. To make the check active again, remove your override or fix it, and request the HealthChecker to refresh the best practices policy.

The HealthChecker reports reflect values at a point in time (snapshot). The report is comprised of a series of records in the Sytem Logger. These records have the following components:

- Message text and explanation
- Recommendations of actions that can be taken to address an exception
- IBM suggestions
- Reasons for IBM's suggestions

The HealthChecker generates two report formats:

- **Regular**

  The regular format produces confirmation messages of those checks where the results meet either IBM's or your override values. For selected checks, such as several related to consoles or UNIX® System Services file systems, additional information about these resources is provided.
- **Exception**

  The exception format provides status on checks that do not meet the criteria for the check.

To help distinguish a successful check from one encountering an exception, the record is explicitly marked in column 'E' as an I(BM) or U(ser) exception. In addition, the status of an exception is explicitly noted in the reports as:

```
*Exception: IBM criteria not met*
```

or

```
*Exception: User criteria not met*
```

Unless you take any of the actions given in "Types of actions" on page 156, the HealthChecker performs regular checks at predefined time intervals. The time intervals are defined individually for each check as part of IBM's best practices and can be overridden. The checks are done based on IBM's best practices or your overrides.

Types of actions

You can request the HealthChecker to run selected checks or all checks at a point in time when you want it to.

You can request the HealthChecker to update or refresh your overrides (see "Customization" on page 22).

Recommendation

You should iteratively run the HealthChecker function and take corrective action until you have no exceptions. You should either update the values used in your environment, override the IBM values, or suppress the IBM check. This is important so that on subsequent runs of the HealthChecker, you will only see exceptions that you should attend to. Otherwise, the reports may contain a mixture of messages that you regularly ignore and those that could reflect a new potential problem, making it more likely that you could miss a key exception message.

Minimally, you should run the HealthChecker when you reIPL. However, many values can change between IPLs. You may therefore want to consider the following:

- permanently having the HealthChecker perform the checks repeatedly based on their respective, defined time interval. If the time intervals that are predefined by IBM do not suit your needs, you can always override them for individual checks using the TIMEINT parameter.
- if you don't want to have the HealthChecker permanently active, you can activate it at regular intervals (for example, once a day) and then deactivate it.

Syntax

INGHC

```
►►──INGHC──┬──────────────────────────┬──┬──────────────────────────────┬──────►
           └──CHECK──=──┤ checklist ├─┘  └──SYSTEMS──=──┤ systemlist ├──┘

►──┬───────────────────────┬──┬─────────────────────┬──────────────────────────►
   └─START──=──timestamp───┘  └─END──=──timestamp───┘

►──┬──────────────────────────────────────┬────────────────────────────────────►◄
   ├─OUTMODE──=──LINE─────────────────────┤
   └─OUTDSN──=──dsname──┬───────────────────┬┘
                        └─OUTMODE──=──LINE──┘
```

checklist:

```
            ┌──────────────────┐
            ▼                  │
├──(──────┬─CDS──────────┬─────┴──)──────────────────────────────────────┤
          ├─CF───────────┤
          ├─CONsoles─────┤
          ├─EXceptions───┤
          ├─Latestcheck──┤
          ├─Other────────┤
          ├─REFResh──────┤
          ├─NEWCheck─────┤
          ├─SYStem───────┤
          ├─XCFRecovery──┤
          └─XCFSIgnal────┘
```

systemlist:

```
├──┬─sysname──────────────────┬──────────────────────────────────────────┤
   │        ┌──────────┐      │
   │        ▼          │      │
   └─(──────sysname────┴──)───┘
```

Parameters

CHECK
Defines the filter options for the data to be displayed (default: the result of all checks):

CDS Retrieves results relating to all of the various checks of all the couple datasets defined in the sysplex.

CF Retrieves results relating to all of the various checks of all coupling facilities and structures in the sysplex.

CONsoles Retrieves results relating to all of the various checks of all consoles defined in the sysplex.

EXceptions Retrieves only results indicating an exception. This means the resource being checked does not meet either the IBM criteria or the user criteria.

SYStem Retrieves results relating to system resource checks.

Latestcheck Retrieves only the latest results of any kind of check.

Other Retrieves results which do not belong to any of the filters.

REFResh Triggers the refresh of the user-defined best practices, an evaluation of the new values, and finally displays the new report data. This includes NEWCheck for all defined checks. This function is only valid when it is executed on a system that is running msys for Operations and, as a consequence, the command ACF COLD is executed implicitly on that system in order to read the user's data from AOFCUST. The request is sent to all systems specified.

NEWCheck Requests the HealthChecker to immediately run checks as specified in the filter options and display the reports for those checks. The request is sent to all specified systems.

XCFREcovery Retrieves results relating to checks of XCF recovery settings.

XCFSIgnal Retrieves results relating to the XCF path checks.

If you specify more than one filter option, they must be enclosed by parentheses, for example, `INGHC CHECK=(EX L)`. When the CHECK parameter is not provided, the DEFAULT will assume ALL checks.

SYSTEMS

Specifies the list of systems for which the HealthChecker results should be retrieved, and where the specified actions are to be performed. If you specify more than one system name, they must be enclosed by parentheses, for example, `INGHC SYSTEMS=(sys1 sys2)`. When the SYSTEM parameter is not provided, the DEFAULT will be all systems in the sysplex.

sysname

The name of the system.

START

Is the start date and, optionally, start time for the display of the history data. The format is *yyyy-mm-dd [hh:mm:ss]*. If omitted, the history data from the last hour will be displayed, even if other parameters (such as check, systems) are specified. If you specify more than one value, they must be enclosed by parentheses, for example `INGHC START=(2002-09-26 08:00:00)`.

END Is the end date and, optionally, end time for the display of the history data. The format is the same as for the START parameter.

timestamp

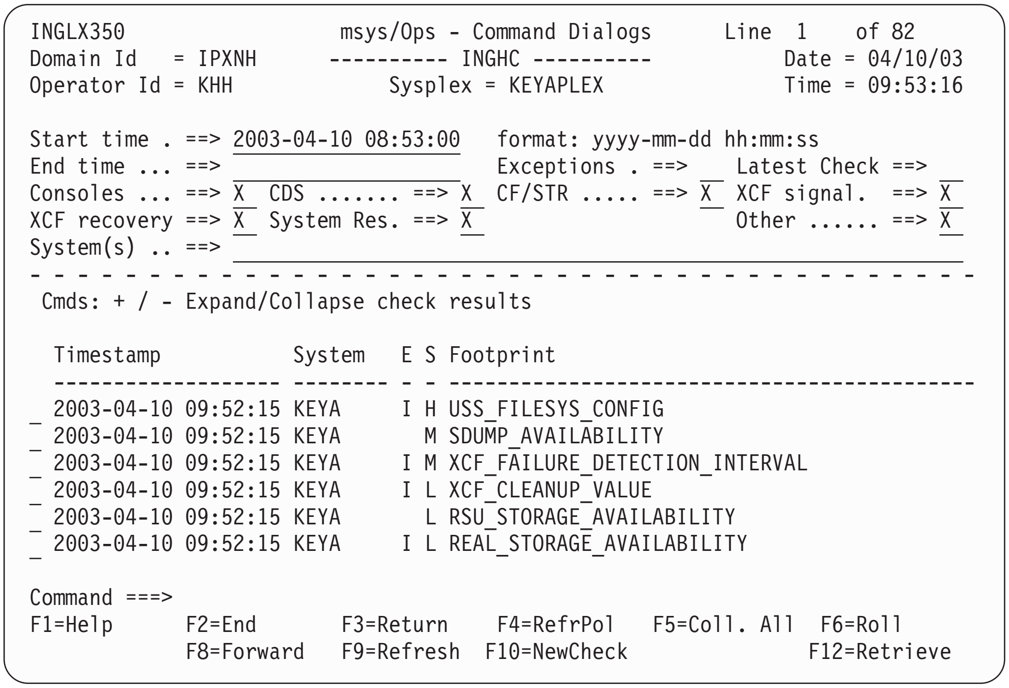
The date and time in the format *YYYY-MM-DD [hh:mm:ss]*.

OUTMODE

If you specify OUTMODE=LINE, INGHC is called in line mode. In this mode, only the display functions of the command are available, as shown in Figure 27 on page 163.

Examples

When you issue the INGHC command, the panel in Figure 24 on page 159 is displayed.



```
INGLX350                  msys/Ops - Command Dialogs      Line  1    of 82
Domain Id   = IPXNH      ---------- INGHC ----------           Date = 04/10/03
Operator Id = KHH             Sysplex = KEYAPLEX               Time = 09:53:16

Start time . ==> 2003-04-10 08:53:00   format: yyyy-mm-dd hh:mm:ss
End time ... ==> ___________________   Exceptions . ==> __  Latest Check ==> __
Consoles ... ==> X  CDS ....... ==> X  CF/STR ..... ==> X  XCF signal.  ==> X
XCF recovery ==> X  System Res. ==> X                       Other ...... ==> X
System(s) .. ==> ______________________________________________________________
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 Cmds: + / - Expand/Collapse check results

  Timestamp           System   E S Footprint
  ------------------- -------- - - ------------------------------------------
_ 2003-04-10 09:52:15 KEYA     I H USS_FILESYS_CONFIG
_ 2003-04-10 09:52:15 KEYA       M SDUMP_AVAILABILITY
_ 2003-04-10 09:52:15 KEYA     I M XCF_FAILURE_DETECTION_INTERVAL
_ 2003-04-10 09:52:15 KEYA     I L XCF_CLEANUP_VALUE
_ 2003-04-10 09:52:15 KEYA       L RSU_STORAGE_AVAILABILITY
_ 2003-04-10 09:52:15 KEYA     I L REAL_STORAGE_AVAILABILITY

Command ===>
F1=Help      F2=End       F3=Return    F4=RefrPol   F5=Coll. All  F6=Roll
             F8=Forward   F9=Refresh  F10=NewCheck              F12=Retrieve
```

Figure 24. INGHC panel

This panel displays the results of all Health Check requests. The fields in the non-scrollable area, in the top portion of the screen, allow filtering of Health Check requests as follows:

Start time
The date and time start filter in YYYY-MM-DD HH:MM:SS format converted to the local timezone. This defaults to the current time less one hour if not provided.

End time
The date and time end filter in YYYY-MM-DD HH:MM:SS format converted to the local timezone. This defaults to the current time if not provided.

Exceptions
The Exceptions filter returns *only* checks that show an exception that occurred between the Start and End times. This display can also be limited by using the other filters in this area.

Latest Check
The Latestcheck filter returns *only* the latest checks that have been performed. This display can also be limited by using the other filters in this area. You may have no output if the Latestcheck was performed at a time outside the Start and End times provided.

System(s)
The System(s) field enables you to restrict data to a specific system or group of systems in a sysplex.

Note: All systems are assumed if this field is left blank.

The following allows you to filter the checks you are interested in. You cannot filter individual checks, however, because the checks are grouped into categories. The following categories have been defined:

Consoles
The Consoles field returns all console-related results.

CDS
The CDS field returns all Couple-Data-Set-related results.

CF/STR
The CF/STR field returns all Coupling-Facility-Structure-related results.

XCF signal. The XCF Signal field returns all XCF-Signalling-related results.

XCF recovery The XCF Recovery field returns all XCF-RECOVERY-related results.

System Res. The System Res. field returns all system-related check results.

Other The Other field returns all data not covered in the previous filters.

The lower portion of the panel is a scrollable area where the results of your requests are displayed. The output fields contain the data in the following format:

Timestamp The Timestamp field indicates the actual date and time that the health check was performed.

System The System field displays the system that the health check was performed on.

E(xception) The Exception field indicates whether there is a deviation from the specified best practices (either IBM or User defined). An 'I' will indicate an IBM exception exists, a 'U' indicates that a User exception exists. A blank indicates that there are no exceptions.

S(everity) The Severity field indicates the importance of the item. Its value is defined in the IBM best practices or user overrides. This field contains either L (Low), M (Medium), H (High).

Footprint The Footprint contains the name of the check item. For the list of all valid names see the online help. To view an item or group of items in more detail, enter a '+' in the field to the left of the item(s) you wish to display and press Enter. To reset the display, enter a '-' in the field to the left of the item and press Enter to collapse it. The amount of detail displayed is limited by the size of one record in the system logger. If the output exceeds this limit, it is truncated, and the respective footprint is displayed in a different color.

Three PF keys are specific to this panel.

RefrPol (PF4) Triggers the refresh of the user-defined best practices, and an evaluation of the new values. This includes NEWCheck for all checks. The request is sent to the specified systems – the default is ALL. This Implicitly calls the ACF COLD command in order to read the user's data from AOFCUST. Afterwards, the report data is displayed, according to the filter options.

Note: If your specification for a certain check is in error, then the check will not be performed until the error is fixed.

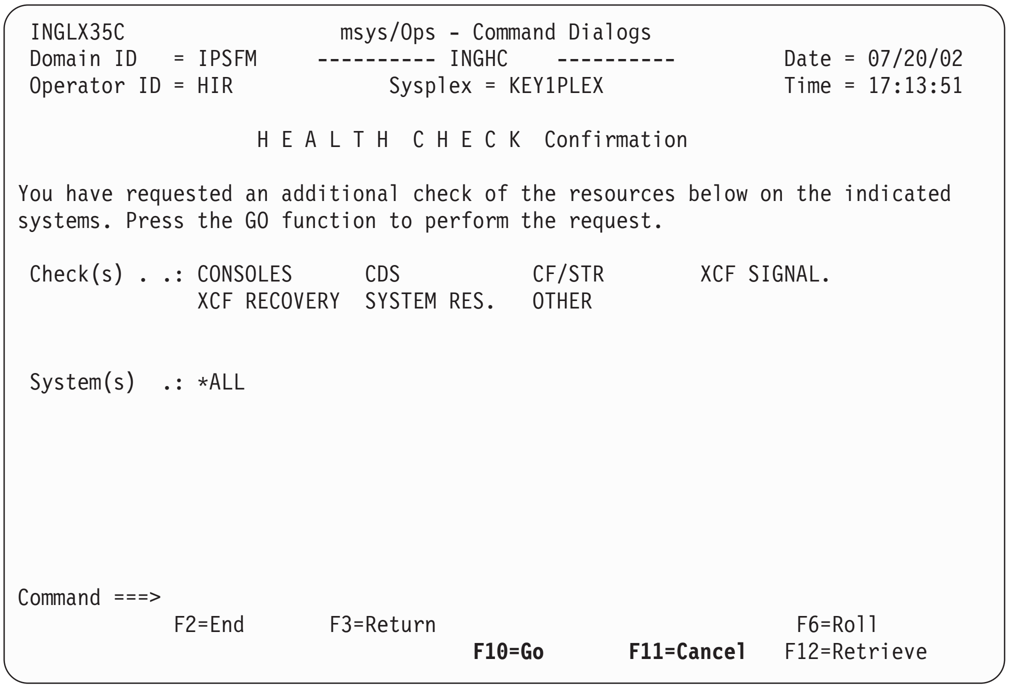
Coll. All (PF5) Collapses the detail description of all check results.

NewCheck (PF10)
Requests the HealthChecker to immediately run checks as specified in the filter options. The request is sent to all specified systems, the default is ALL. The report data is displayed, according to the filter options.

Note: Even if the input fields of the filter options show only an 'X' when selected they must be 2 characters long to support up to 26 footprint lines due to NetView restrictions.

Note: You should check that the systems listed in the Policy Refresh and New Check confirmation panels are those for which you want to refresh the user-defined best practices or perform a new check. This is because PF keys work such that they do not accept new values in input fields. Instead, they reset the input fields to the values last entered with the Enter key and then run the requested function. If RefrPol or Newcheck are wanted for another system or for some other filter in general (systems are a filter criterion) you first need to change those values and press Enter. The new values are then validated for correctness, and then the respective PF key can be executed correctly. For more information, see Chapter 10, “Moving between the components and using function keys,” on page 111.

Pressing the PF10 key to request a NewCheck displays the Health Check confirmation screen, as shown in Figure 25.



```
 INGLX35C                 msys/Ops - Command Dialogs
 Domain ID   = IPSFM     ---------- INGHC    ----------         Date = 07/20/02
 Operator ID = HIR             Sysplex = KEY1PLEX               Time = 17:13:51

                   H E A L T H  C H E C K  Confirmation

You have requested an additional check of the resources below on the indicated
systems. Press the GO function to perform the request.

 Check(s) . .: CONSOLES      CDS           CF/STR        XCF SIGNAL.
               XCF RECOVERY  SYSTEM RES.   OTHER


 System(s)  .: *ALL









Command ===>
             F2=End       F3=Return                              F6=Roll
                                     F10=Go       F11=Cancel   F12=Retrieve
```

Figure 25. INGHC check confirmation panel

This panel shows the selected check criteria to be performed and the systems involved. If you want to cancel the new check, press PF11, otherwise press PF10 to perform the request.

The output, when returned, will be sorted in descending Timestamp order, as shown in Figure 24 on page 159. You may sort the output by issuing the SORT command on the command line, followed by the sort direction (ascending or descending) and the column to sort. For example, to sort on Severity in descending order issue:

```
sort d 4
```

To search for text strings, issue the FIND, F, RFIND or RF commands. For example, to find the string *consoles*, enter the following:

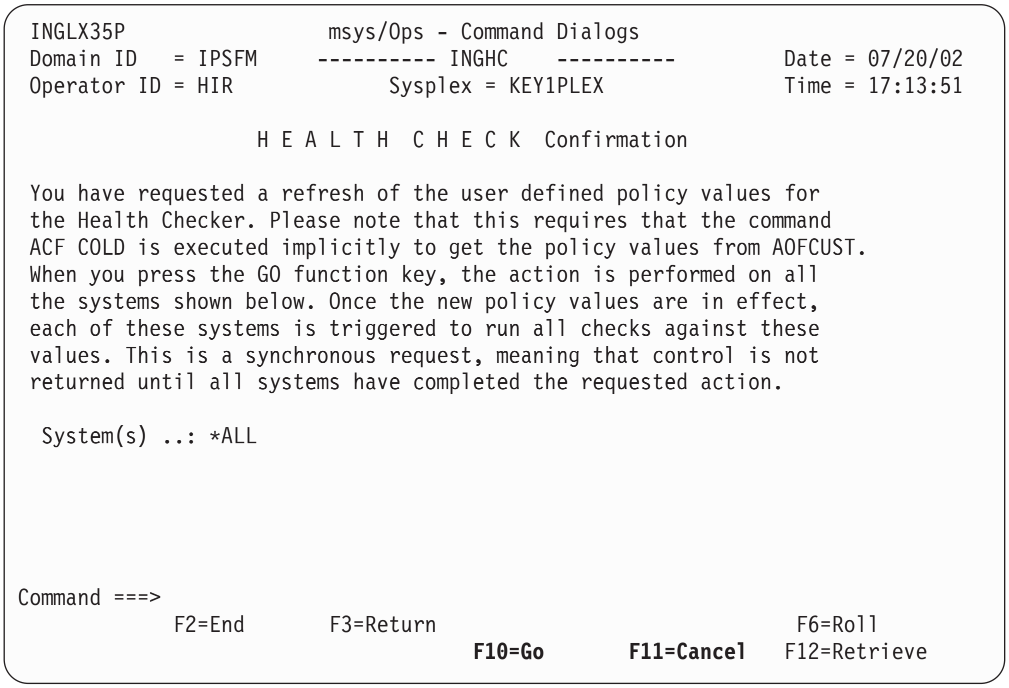
```
f consoles n
```

Note: The FIND command can search *only* the text for those check results that have been expanded, even when the text is not visible on the action screen.

1 For more information about sorting output and searching for text strings, refer to
1 the online help.

1 To view an item or group of items in more detail, enter a '+' in the field to the left
1 of the item(s) you wish to display. To reset the display, enter a '-' in the field to the
1 left of the item to 'collapse', or use PF5 to collapse all expanded items.

1 To refresh the user-defined best practices and evaluate the new values, press the
1 PF4 key. The panel shown in Figure 26 is displayed.
1



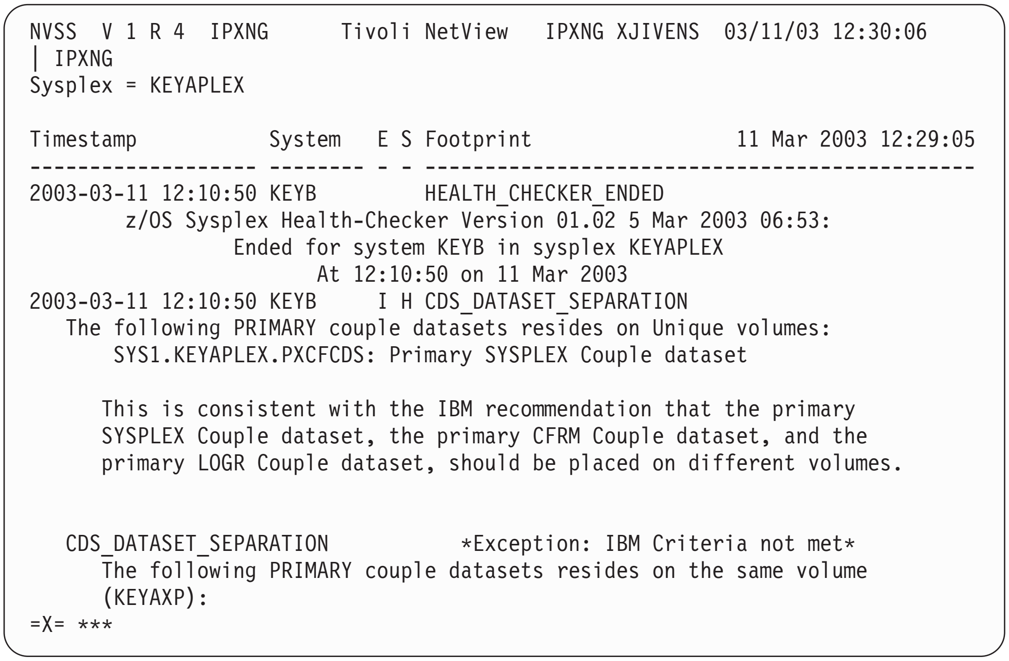
```
 INGLX35P                msys/Ops - Command Dialogs
 Domain ID   = IPSFM     ---------- INGHC    ----------         Date = 07/20/02
 Operator ID = HIR            Sysplex = KEY1PLEX                Time = 17:13:51

                   H E A L T H  C H E C K  Confirmation

 You have requested a refresh of the user defined policy values for
 the Health Checker. Please note that this requires that the command
 ACF COLD is executed implicitly to get the policy values from AOFCUST.
 When you press the GO function key, the action is performed on all
 the systems shown below. Once the new policy values are in effect,
 each of these systems is triggered to run all checks against these
 values. This is a synchronous request, meaning that control is not
 returned until all systems have completed the requested action.

  System(s) ..: *ALL





Command ===>
             F2=End       F3=Return                              F6=Roll
                                      F10=Go       F11=Cancel   F12=Retrieve
```

Figure 26. INGHC refresh confirmation panel

1 If you want to cancel the policy refresh, press PF11, otherwise press PF10 to
1 perform the request.

1 Line mode output

1 If you issue the INGHC command with OUTMODE=LINE, the output shows both
1 the footprint data as well as the detail data, and will be similar to that shown in
1 Figure 27 on page 163.
1



```
NVSS  V 1 R 4  IPXNG      Tivoli NetView   IPXNG XJIVENS  03/11/03 12:30:06
| IPXNG
Sysplex = KEYAPLEX

Timestamp           System   E S Footprint                11 Mar 2003 12:29:05
------------------- -------- - - ---------------------------------------------
2003-03-11 12:10:50 KEYB         HEALTH_CHECKER_ENDED
        z/OS Sysplex Health-Checker Version 01.02 5 Mar 2003 06:53:
                 Ended for system KEYB in sysplex KEYAPLEX
                        At 12:10:50 on 11 Mar 2003
2003-03-11 12:10:50 KEYB     I H CDS_DATASET_SEPARATION
   The following PRIMARY couple datasets resides on Unique volumes:
       SYS1.KEYAPLEX.PXCFCDS: Primary SYSPLEX Couple dataset

      This is consistent with the IBM recommendation that the primary
      SYSPLEX Couple dataset, the primary CFRM Couple dataset, and the
      primary LOGR Couple dataset, should be placed on different volumes.


   CDS_DATASET_SEPARATION           *Exception: IBM Criteria not met*
      The following PRIMARY couple datasets resides on the same volume
      (KEYAXP):
=X= ***
```

Figure 27. INGHC line mode output

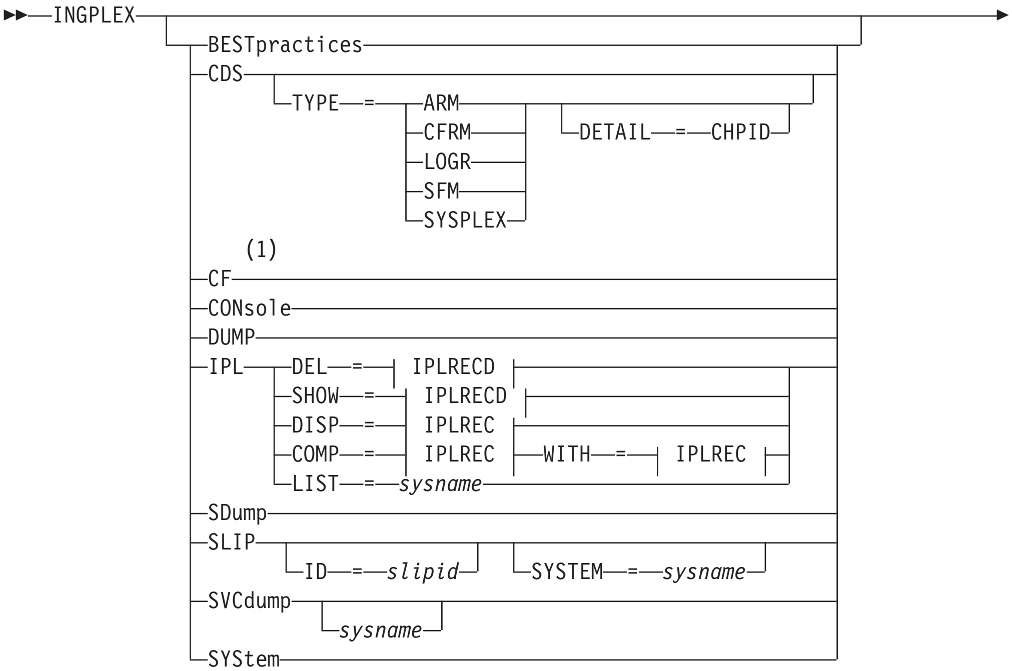
1
1 INGPLEX

1 Purpose

1 The INGPLEX command comprises all the sysplex-related functions of msys for
1 Operations. It can be called in full mode and in line mode. If it is called in line
1 mode, only the display functions are available. Therefore, you cannot start an
1 action in msys for Operations if you issue INGPLEX from an MVS console.

1 Syntax

1



```
►─┬─────────────┬──────────────────────────────────────────────────────►◄
  └─OUTMODE=LINE─┘
```

IPLREC:

```
├──sysname──/──timestamp──[/──member──[/──suffix──]]──────────────────────┤
```

IPLRECD:

```
├──sysname──/──timestamp──────────────────────────────────────────────────┤
```

Notes:

1 For details see "INGCF" on page 134.

Parameters

1 **BESTpractices**
1 Displays information about the currently active HealthChecker best practices.
1 This information is retrieved from the system that performs the global and
1 local checks.

CDS

Displays information about CDSs and supports replacement of the current alternate CDS by a new one as well as making the alternate CDS the new primary. For further information about INGPLEX CDS refer to "CDS" on page 169.

TYPE

The type of CDS for which the CDS function is issued. Possible values are ARM, CFRM, LOGR, SFM, and SYSPLEX.

DETAIL

If you specify this parameter with the CDS function, the channel paths for the respective CDS type are displayed.

CF

This is the equivalent of the INGCF command.

CONsole

Displays information about consoles.

DUMP

Shows the DUMP submenu.

IPL

Shows and compares IPL information. It can be issued with the following options:

DEL

Deletes a single IPL record and all its related information. Note that the DEL parameter is supported in line mode only.

SHOW

Shows the details panel of the specified IPL record.

DISP

Shows all, one, or particular PARMLIB members used by the IPL of the specified system and at the specified date and time.

COMP
Compares all, one, or particular PARMLIB members used by the IPL of the specified system and at the specified date and time with those specified in the WITH parameter.

LIST
Shows the IPL summary records of the specified system.

WITH
The COMP parameter compares all, one, or particular PARMLIB members used by the IPL of the specified system, at the specified date and time with those specified in parameter WITH.

sysname
Is the name of the system in the sysplex.

timestamp
Is the IPL date and time. The format is YYYYMMDDhhmm.

member
Is the name of the PARMLIB member without the suffix.

suffix
Is the suffix of the PARMLIB member.

For further information about INGPLEX IPL refer to "IPL" on page 179.

SDUMP
Displays and controls the SDUMP options being set on all systems in the sysplex. For further information about INGPLEX SDUMP refer to "SDUMP" on page 181.

SLIP
Displays and controls all SLIP traps of all systems in the sysplex. Controlling is limited to DISABLE, ENABLE, or REMOVE a SLIP trap. The following parameters are supported:

ID Limits the line mode output and the initial full screen display to the particular SLIP trap ID

slipid
Is the ID of a SLIP trap. It can consist of one to four characters. Wildcards are not supported.

SYSTEM
Limits the line mode output and the initial full screen display to the particular system.

For further information about INGPLEX SLIP refer to "SLIP" on page 186.

SVCDUMP
Allows you to issue a multisystem dump of up to 15 address spaces including data spaces owned by the address spaces, structures used by the address spaces, and XCF group members on the same or on other systems in the sysplex of those groups the address spaces have joined. The following parameter is supported:

sysname
Is the name of the system having joined the XCF group of the NetView the operator is logged on to.

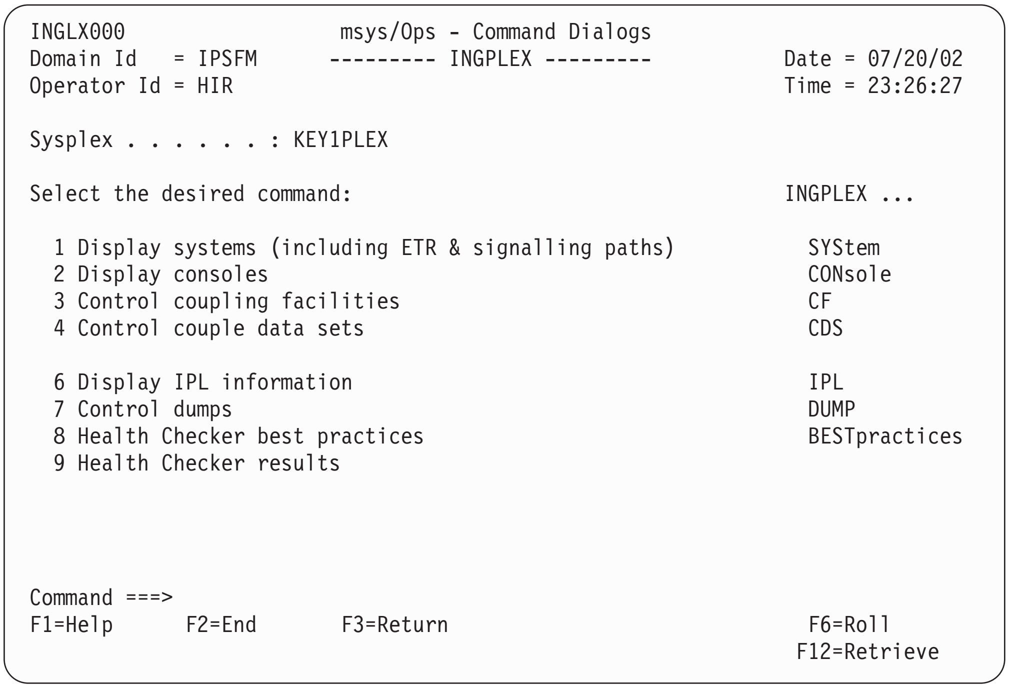
For further information about INGPLEX SVCDUMP refer to “SVCdump” on page 183.

SYStem
Displays information about a member system of the sysplex.

OUTMODE
If you specify OUTMODE=LINE, INGPLEX is called in linemode. In this mode, only the display functions of the command are available.

Example

If you specify INGPLEX without parameters, the following selection panel is displayed:



```
INGLX000                  msys/Ops - Command Dialogs
Domain Id   = IPSFM      --------- INGPLEX ---------           Date = 07/20/02
Operator Id = HIR                                              Time = 23:26:27

Sysplex . . . . . . . : KEY1PLEX

Select the desired command:                                    INGPLEX ...

  1 Display systems (including ETR & signalling paths)           SYStem
  2 Display consoles                                             CONsole
  3 Control coupling facilities                                  CF
  4 Control couple data sets                                     CDS

  6 Display IPL information                                      IPL
  7 Control dumps                                                DUMP
  8 Health Checker best practices                                BESTpractices
  9 Health Checker results





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

Figure 28. INGPLEX selection panel

Specify the number or the function and press Enter.

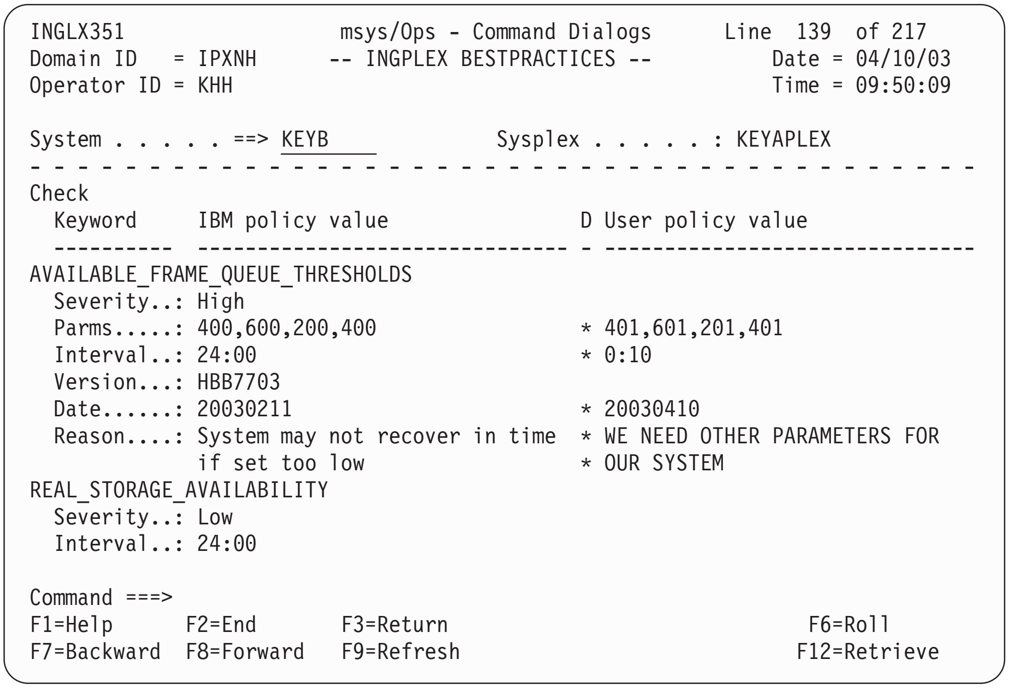
BESTpractices

Purpose

This command allows you to view the currently active best practices from the system doing global checks, which is shown on the panel.

Example

When you issue the INGPLEX BEST command, to display the IBM and User-defined best practices, the panel in Figure 29 on page 167 is displayed.



```
INGLX351                 msys/Ops - Command Dialogs      Line  139  of 217
Domain ID   = IPXNH      -- INGPLEX BESTPRACTICES --         Date = 04/10/03
Operator ID = KHH                                            Time = 09:50:09

System . . . . . ==> KEYB____          Sysplex . . . . . : KEYAPLEX
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Check
  Keyword     IBM policy value               D User policy value
  ----------  ------------------------------ - ------------------------------
AVAILABLE_FRAME_QUEUE_THRESHOLDS
  Severity..: High
  Parms.....: 400,600,200,400                * 401,601,201,401
  Interval..: 24:00                          * 0:10
  Version...: HBB7703
  Date......: 20030211                       * 20030410
  Reason....: System may not recover in time * WE NEED OTHER PARAMETERS FOR
              if set too low                 * OUR SYSTEM
REAL_STORAGE_AVAILABILITY
  Severity..: Low
  Interval..: 24:00

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

Figure 29. INGPLEX BESTpractices panel

This panel displays the IBM's and user best practices of the HealthChecker function. The following fields are shown in the non-scrollable area of the screen.

System
Name of the system from which the best practices are retrieved.

The best practices can only be retrieved from a system where the HealthChecker function is active. If multiple systems are eligible, the system where the checks with sysplex scope (global checks) are done is chosen by default.

You may choose another system by overtyping the system name with the name of the desired system. If the system name is cleared, the best practices are retrieved from the default system.

Sysplex
Name of the sysplex that the system named above is part of.

The scrollable area of the screen lists the IBM best practice recommendation and, if defined, the user override value.

Check
Best practice checkname is displayed in white.

Keyword
Associated keywords for a check are displayed below the checkname.

IBM policy value
Details the IBM Bestpractice value. The following values are defined:

Severity
Expresses a sense of urgency about the need to fix the exceptional situation for the check. Possible values are High, Medium, Low.

Parms
IBM's recommended settings for the check.

Interval
IBM's recommendation of how often to perform the check. The time interval is shown in the format *hh:mm*.

Version
Shows the FMID of the operating system release for which the check is applicable. This can be a range.

Date
Shows the date when IBM introduced or last modified the check.

Reason
Documents the reason why the check should be done.

D
An asterix in this column represents a discrepancy between the IBM recommended value and a User override.

User policy value
Details the user override values. The following values can be overridden:

- Parms, where applicable
- Interval
- Date
- Reason
- Severity

The value for Interval is also used to mark checks which are not performed on the system for the following reasons:

NOCALL specified
Indicates that in the user policy the check is specified explicitly with NOCALL.

NOCALL - Check failed
Indicates the checker failed 3 times.

NOCALL - System error
Indicates an unrecoverable system error in this check. It might also indicate an error in the IBM Parms.

NOCALL - User Error
Indicates an error in the user parms for this check.

NOCALL - n/a
Indicates the check is not applicable on the system. For example if the system is set up in Monoplex mode, certain checks are not applicable.

NOCALL - Global check n/a
Indicates that the check is not executed because some other system performs the global checks.

NOCALL - ?????
Indicates that the check is not executed for some undefined reason.

When a User value does not match an IBM recommended value, along with the discrepancy flag, the user value is displayed in a different color.

The display of a particular keyword is suppressed if it is not applicable to the check.

CDS

Purpose

The CDS function displays information about all the couple data sets in the system, including details of the corresponding policies. For every CDS type that is required by the implementation INGPLEX CDS allows the operator to:

- Switch from the primary to the alternate CDS
- Define a new alternate CDS
- Change the active policy (if applicable)

Actions are started by specifying an action code for a selected CDS type on the panel.

Actions

The possible action codes are:

***allocate alternate CDS (A)**
Replaces the current alternate CDS for a selected CDS type with a new one. There are two options how to do this:

- The alternate CDS is allocated automatically by msys for Operations.

  This automatic allocation requires that spare volumes have been defined in AOFCUST, and that one of these spare volumes is available.
- Specify the data set that is to be used as the new alternate CDS.

  If you specify your own data set, observe the following:
  - The data set must exist
  - It must have been formatted with the XCF formatting tool
  - It must be at least as large as the current primary CDS, which means that every value you have passed to the XCF formatting tool (for example, in the case of a sysplex CDS, the maximum number of systems supported) must be equal to or greater than the corresponding value of the primary CDS.

display CHPIDs (C)
Displays information about the channel paths for the selected CDS type.

display CDS information (D)
Displays detail information about the selected CDS type. This comprises the formatting parameters and the policies that are contained in the CDS, if applicable. When the CDSs of the selected type contain policies, the detail information panel provides further actions, namely:

display policy (D)
Displays details about the selected policy.

***start policy (S)**
Makes the selected policy the active policy.

The policy switch must be confirmed before it is executed.

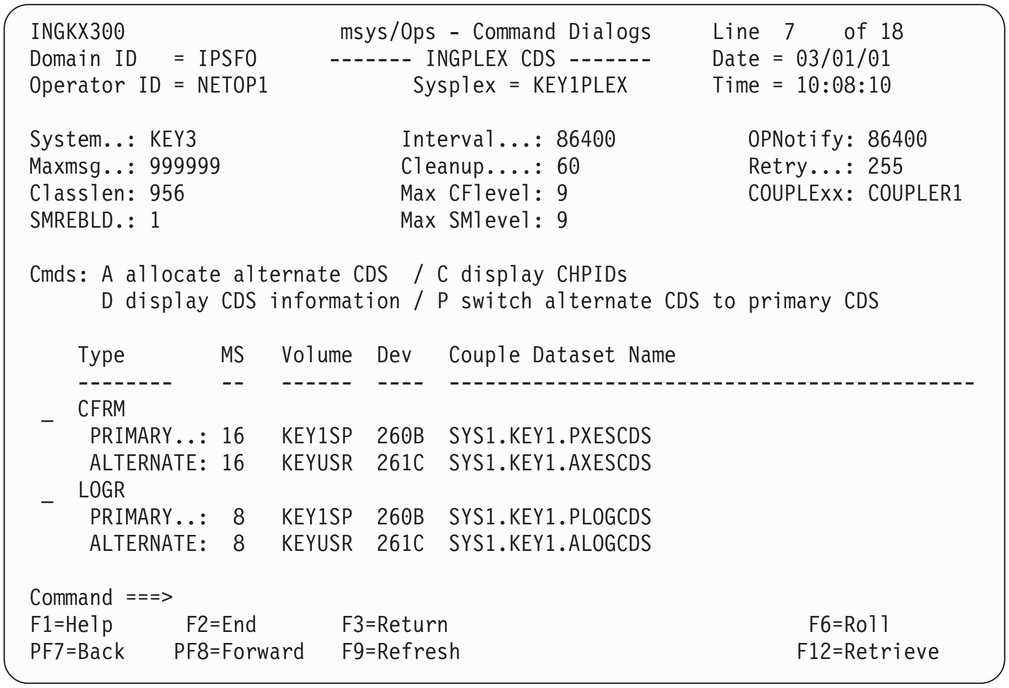
***switch alternate CDS to primary CDS (P)**
Makes the alternate CDS the primary one. Because an alternate CDS is no longer available after the switch, msys for Operations shows a confirmation panel before the action is performed. On the panel, you can specify a new alternate CDS. When CDS recovery is switched on and you do not supply your own alternate CDS, msys for Operations tries to allocate a new alternate CDS automatically. The special requirements for

manual and automatic creation of the new alternate CDS are the same as those for the replacement of the alternate CDS (action code A).

Examples

The following example illustrates the switch from the primary to the alternate CDS.

The following examples start with issuing `INGPLEX CDS` and pressing F8 on the CDS command dialog to scroll down the CDS list. The following panel is displayed:



```
INGKX300                  msys/Ops - Command Dialogs     Line  7    of 18
Domain ID   = IPSFO      ------- INGPLEX CDS -------     Date = 03/01/01
Operator ID = NETOP1           Sysplex = KEY1PLEX        Time = 10:08:10

System..: KEY3                 Interval...: 86400          OPNotify: 86400
Maxmsg..: 999999               Cleanup....: 60             Retry...: 255
Classlen: 956                  Max CFlevel: 9              COUPLExx: COUPLER1
SMREBLD.: 1                    Max SMlevel: 9

Cmds: A allocate alternate CDS  / C display CHPIDs
      D display CDS information / P switch alternate CDS to primary CDS

    Type        MS   Volume  Dev   Couple Dataset Name
    --------    --   ------  ----  --------------------------------------------
 _  CFRM
     PRIMARY..: 16   KEY1SP  260B  SYS1.KEY1.PXESCDS
     ALTERNATE: 16   KEYUSR  261C  SYS1.KEY1.AXESCDS
 _  LOGR
     PRIMARY..:  8   KEY1SP  260B  SYS1.KEY1.PLOGCDS
     ALTERNATE:  8   KEYUSR  261C  SYS1.KEY1.ALOGCDS

Command ===>
F1=Help      F2=End       F3=Return                              F6=Roll
PF7=Back    PF8=Forward   F9=Refresh                            F12=Retrieve
```

Figure 30. INGPLEX CDS command dialog panel

The panel header contains sysplex-related information about the system on which the INGPLEX command was executed. The details are as follows:

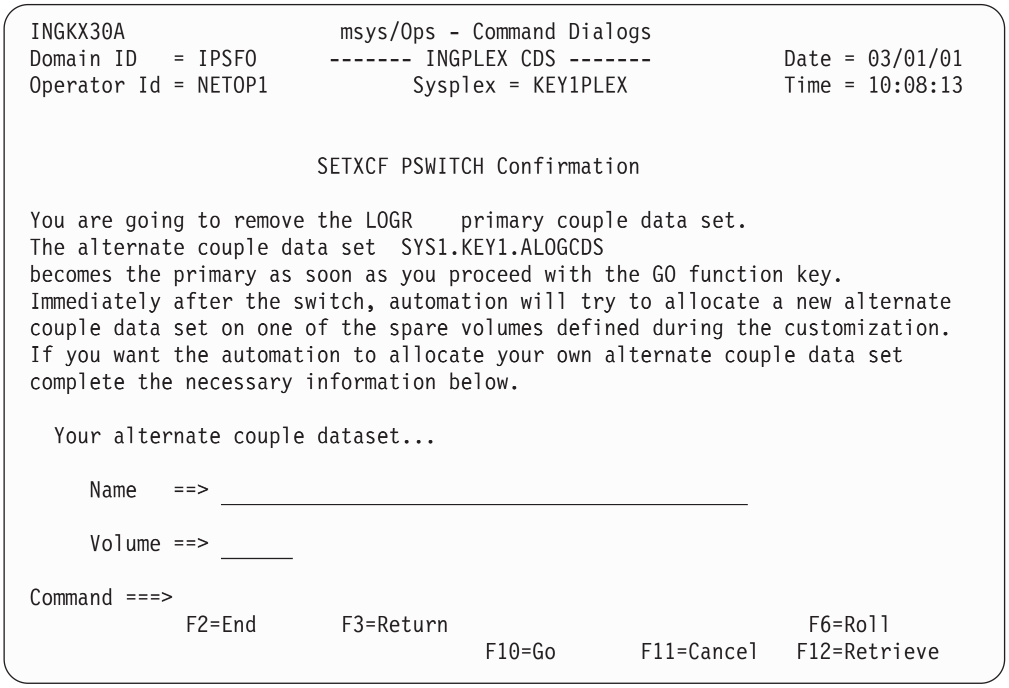
- The **System** field shows the name of the system.
- The **Interval** field shows the system failure detection interval in seconds. This interval is the amount of time XCF lets elapse without a status update before assuming that the system failed.
- The **OPNotify** field shows the number of seconds that XCF waits before notifying the operator of a potential system problem.
- The **Maxmsg** field shows the default value for the maximum amount of kilobytes of message buffer space. This default value is used when MAXMSG is not specified on SETXCF START commands.
- The **Cleanup** field shows the number of seconds that XCF waits for cleanup of members.
- The **Retry** field shows the default value for the retry limit. This value is used when the RETRY keyword is not specified on SETXCF START commands.
- The **Classlen** field shows the default length (in bytes) of messages allowed for a transport class. This value is used when CLASSLEN is not specified on the SETXCF START CLASSDEF command.

- The **Max CFlevel** field shows the maximum CFLEVEL supported by this system. This system can connect to a coupling facility with a higher CFLEVEL than the value of **Max CFlevel** but would not be enabled to use any functions supported by the higher level coupling facility.
- The **COUPLExx** field shows the COUPLE*xx* Parmlib member used for system IPL.
- The **SMRBLD** field shows whether (value 1) or not (value 0) system-managed rebuild has been activated in the CFRM couple dat set.
- The **Max SMlevel** field shows the maximum system-managed process level supported by this system.

The main part of the screen shows information about the primary and alternate CDSs for every CDS type. Press F8 to scroll and display further entries. The **MS** column shows the maximum number of systems that are supported by the CDS.

Making an alternate CDS the primary CDS: In this example, the alternate LOGR couple data set is made the new primary CDS. A new alternate CDS is automatically generated.

To switch the LOGR couple data set, enter `P` before `LOGR` on the panel displayed in Figure 30 on page 170, and press ENTER. INGPLEX CDS displays the following confirmation panel:

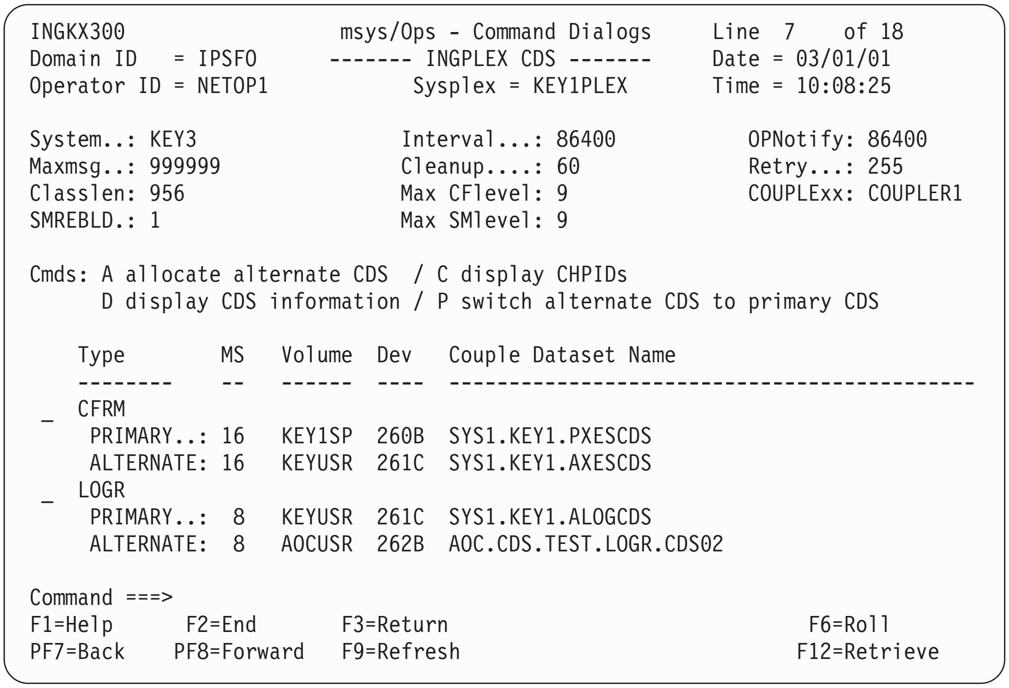


```
INGKX30A                 msys/Ops - Command Dialogs
Domain ID   = IPSFO     ------- INGPLEX CDS -------          Date = 03/01/01
Operator Id = NETOP1           Sysplex = KEY1PLEX            Time = 10:08:13


                         SETXCF PSWITCH Confirmation

You are going to remove the LOGR    primary couple data set.
The alternate couple data set  SYS1.KEY1.ALOGCDS
becomes the primary as soon as you proceed with the GO function key.
Immediately after the switch, automation will try to allocate a new alternate
couple data set on one of the spare volumes defined during the customization.
If you want the automation to allocate your own alternate couple data set
complete the necessary information below.

  Your alternate couple dataset...

     Name   ==> ____________________________________________

     Volume ==> ______

Command ===>
             F2=End       F3=Return                              F6=Roll
                                      F10=Go       F11=Cancel   F12=Retrieve
```

Figure 31. Confirmation panel for switching from the current primary CDS to the alternate CDS

Use this panel to determine how a new alternate CDS is to be created after the switch. You can either specify your own new alternate CDS or let msys for Operations create it for you. When you specify the new alternate CDS yourself, the data set must exist and must have been formatted with the XCF formatting tool. Automatic creation requires that spare volumes have been defined for LOGR couple data sets in the AOFCUST customization file.

Pressing F10 causes msys for Operations to generate the new alternate CDS. After returning to the CDS command dialog, refreshing the panel, and scrolling down with F8, the panel looks as follows:



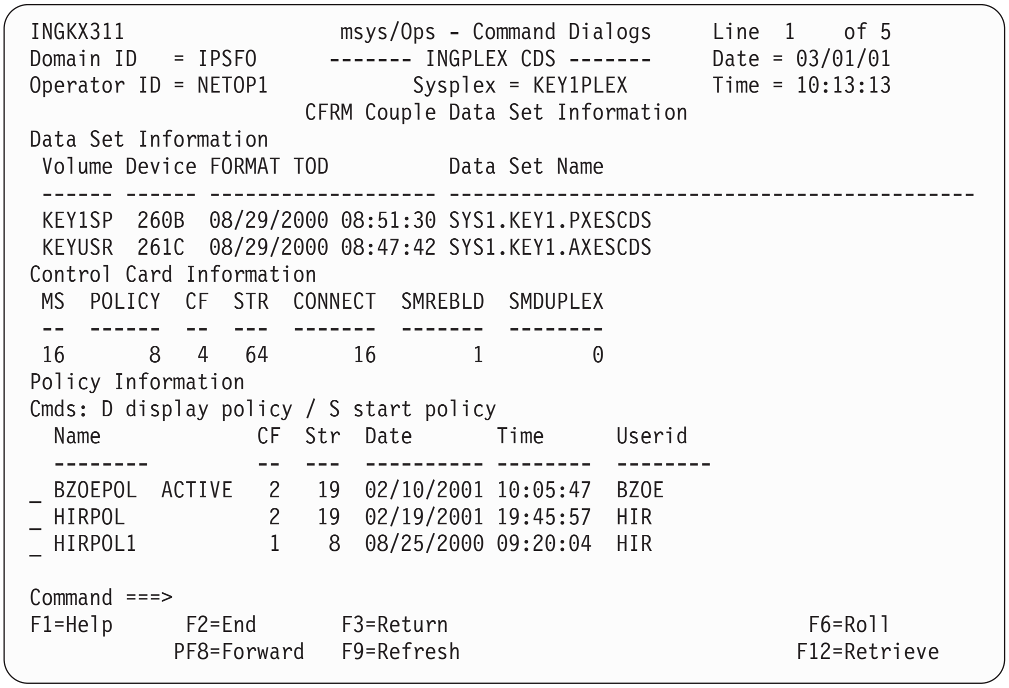
```
INGKX300                  msys/Ops - Command Dialogs     Line  7    of 18
Domain ID   = IPSFO      ------- INGPLEX CDS -------     Date = 03/01/01
Operator ID = NETOP1           Sysplex = KEY1PLEX        Time = 10:08:25

System..: KEY3                 Interval...: 86400          OPNotify: 86400
Maxmsg..: 999999               Cleanup....: 60             Retry...: 255
Classlen: 956                  Max CFlevel: 9              COUPLExx: COUPLER1
SMREBLD.: 1                    Max SMlevel: 9

Cmds: A allocate alternate CDS  / C display CHPIDs
      D display CDS information / P switch alternate CDS to primary CDS

    Type        MS   Volume  Dev   Couple Dataset Name
    --------    --   ------  ----  --------------------------------------------
 _  CFRM
     PRIMARY..: 16   KEY1SP  260B  SYS1.KEY1.PXESCDS
     ALTERNATE: 16   KEYUSR  261C  SYS1.KEY1.AXESCDS
 _  LOGR
     PRIMARY..:  8   KEYUSR  261C  SYS1.KEY1.ALOGCDS
     ALTERNATE:  8   AOCUSR  262B  AOC.CDS.TEST.LOGR.CDS02

Command ===>
F1=Help      F2=End       F3=Return                              F6=Roll
PF7=Back    PF8=Forward   F9=Refresh                            F12=Retrieve
```

Figure 32. INGPLEX CDS command dialog panel after the switch

The previous alternate LOGR CDS has become the primary, and there is a new alternate, which was created by msys for Operations according to the specifications in the CDS section of AOFCUST.

Switching the CFRM policy: In this example, the active CFRM policy is switched.

Enter D before CFRM on the panel displayed in Figure 30 on page 170, and press ENTER. The following panel is displayed:

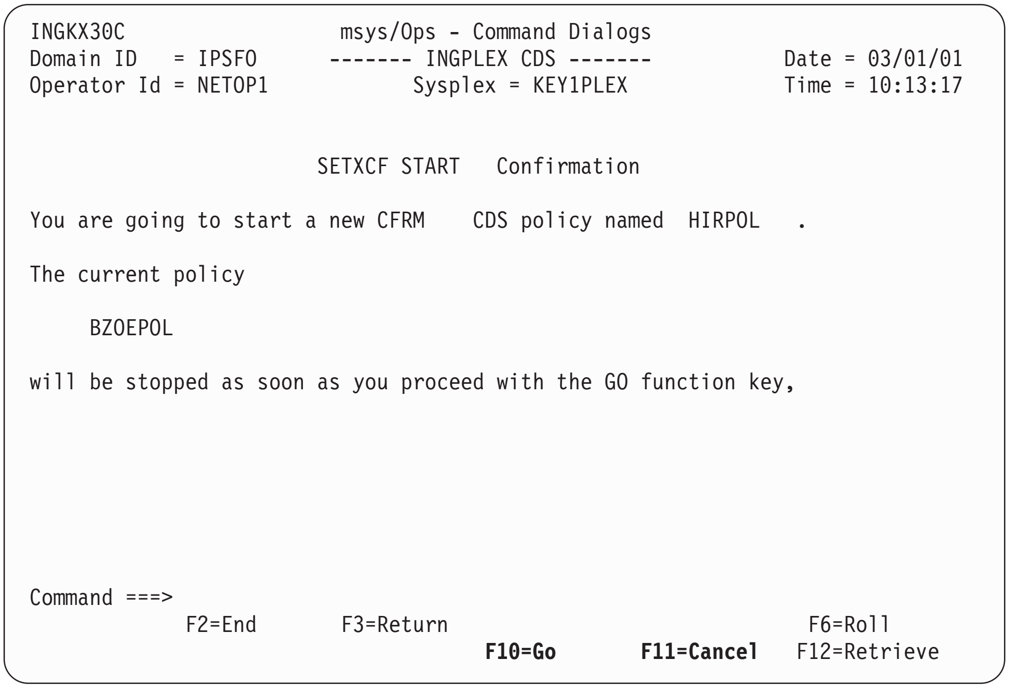


```
INGKX311                  msys/Ops - Command Dialogs     Line  1    of 5
Domain ID   = IPSFO      ------- INGPLEX CDS -------     Date = 03/01/01
Operator ID = NETOP1            Sysplex = KEY1PLEX       Time = 10:13:13
                       CFRM Couple Data Set Information
Data Set Information
 Volume Device FORMAT TOD          Data Set Name
 ------ ------ ------------------- --------------------------------------------
 KEY1SP  260B  08/29/2000 08:51:30 SYS1.KEY1.PXESCDS
 KEYUSR  261C  08/29/2000 08:47:42 SYS1.KEY1.AXESCDS
Control Card Information
 MS  POLICY  CF  STR  CONNECT  SMREBLD  SMDUPLEX
 --  ------  --  ---  -------  -------  --------
 16       8   4   64       16        1         0
Policy Information
Cmds: D display policy / S start policy
  Name             CF  Str  Date       Time      Userid
  --------         --  ---  ---------- --------  --------
_ BZOEPOL  ACTIVE   2   19  02/10/2001 10:05:47  BZOE
_ HIRPOL            2   19  02/19/2001 19:45:57  HIR
_ HIRPOL1           1    8  08/25/2000 09:20:04  HIR

Command ===>
F1=Help      F2=End       F3=Return                              F6=Roll
            PF8=Forward   F9=Refresh                            F12=Retrieve
```

Figure 33. CFRM couple data set information panel before policy switch

The panel shows information about the names and locations of the CDSs. The panel also shows the parameters that were used by the formatting tool of XCF for the allocation of the CDS. The **POLICY** field, for example, displays the maximum number of policies the CDS can contain. Furthermore, the panel shows information about the policies in the CDS, for example, how many coupling facilities and structures are defined in every policy, and which policy is currently active.

To switch to the HIRPOL policy, enter S before this policy and press ENTER. INGPLEX CDS displays the following confirmation panel:

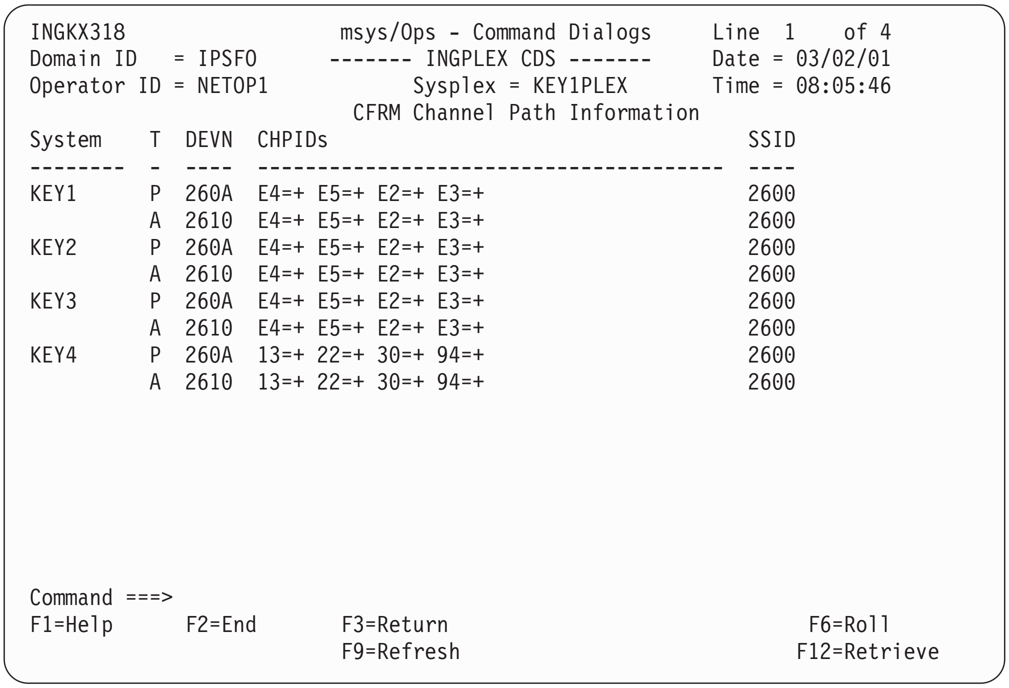


```
INGKX30C                  msys/Ops - Command Dialogs
Domain ID   = IPSFO      ------- INGPLEX CDS -------           Date = 03/01/01
Operator Id = NETOP1            Sysplex = KEY1PLEX             Time = 10:13:17


                        SETXCF START   Confirmation

You are going to start a new CFRM    CDS policy named  HIRPOL   .

The current policy

     BZOEPOL

will be stopped as soon as you proceed with the GO function key,







Command ===>
             F2=End       F3=Return                              F6=Roll
                                      F10=Go       F11=Cancel   F12=Retrieve
```

Figure 34. Confirmation panel for policy switch

Displaying the channel paths for a CDS type: In this example, the channel paths for the CFRM couple data sets are displayed.

Enter `C` before `CFRM` on the panel displayed in Figure 30 on page 170, and press ENTER. The following panel is displayed:



```
INGKX318                  msys/Ops - Command Dialogs     Line  1    of 4
Domain ID   = IPSFO      ------- INGPLEX CDS -------     Date = 03/02/01
Operator ID = NETOP1            Sysplex = KEY1PLEX       Time = 08:05:46
                           CFRM Channel Path Information
System    T  DEVN  CHPIDs                                  SSID
--------  -  ----  ---------------------------------------  ----
KEY1      P  260A  E4=+ E5=+ E2=+ E3=+                      2600
          A  2610  E4=+ E5=+ E2=+ E3=+                      2600
KEY2      P  260A  E4=+ E5=+ E2=+ E3=+                      2600
          A  2610  E4=+ E5=+ E2=+ E3=+                      2600
KEY3      P  260A  E4=+ E5=+ E2=+ E3=+                      2600
          A  2610  E4=+ E5=+ E2=+ E3=+                      2600
KEY4      P  260A  13=+ 22=+ 30=+ 94=+                      2600
          A  2610  13=+ 22=+ 30=+ 94=+                      2600








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

Figure 35. Channel path information for CFRM couple data sets

- The System column shows the name of the sysplex members.

- The T column (for ′type′) indicates whether the CDS is the primary (value ′P′) or alternate (value ′A′).
- The DEVN displays the number of the device on which the CDS resides.
- The CHPIDs column shows the status of the paths to the devices in the format `chpid=status_code`. The codes are those of the operating system. They have the following meaning:

  + The path is logically and physically available and I/O on the path was successful.

  ***** The path is physically, but not logically available. The subchannel’s logical path indicator is off but I/O to the path is successful. You can use the command `VARY PATH (ddd,nn),ONLINE` to make channel path *nn* logically available to device *ddd*.

  – The path is neither logically nor physically available. The subchannel’s logical and physical indicators are both off for this channel path. You can use the command `CONFIG CHP(nn),ONLINE` to make the channel path logically available to all devices connected to the channel.

  & The device is reserved to another path. This indicator applies to devices with the dynamic pathing selection feature.

  < The path is installed but not physically available. The start subchannel request received a condition code of 3.

  > The device microcode has detected an error and will not allow I/O to complete on the path.

  B The path is unable to communicate. The device indicates that a busy or reserve condition exists on the path.

  C A controller error occurred while accessing the device.

  D A device error occurred while accessing the device.

  I Intervention is required; the device is not ready.

  R The path is available and the device is reserved to this path/group. This only applies to devices with the dynamic pathing feature.

  T A time out has occurred; there is no response from the device. The cause of the time out is undetermined and this condition is transient.

  U A storage control unit or storage director error occurred while accessing the device.

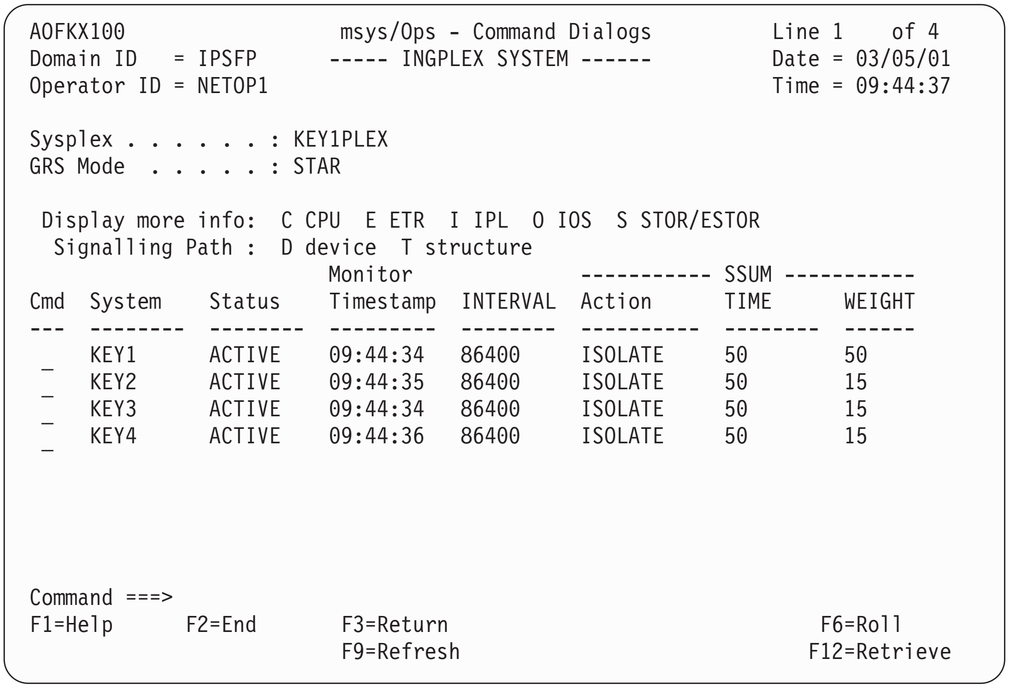
  X Unable to determine the failing unit.
- The SSID field displays the storage subsystem to which the device belongs.

SYStem

Purpose

The SYSTEM function displays the target sysplex name, its GRS mode and its member systems.

Example



```
AOFKX100                  msys/Ops - Command Dialogs          Line 1    of 4
Domain ID   = IPSFP      ----- INGPLEX SYSTEM ------          Date = 03/05/01
Operator ID = NETOP1                                          Time = 09:44:37

Sysplex . . . . . . : KEY1PLEX
GRS Mode  . . . . . : STAR

 Display more info:  C CPU  E ETR  I IPL  O IOS  S STOR/ESTOR
  Signalling Path :  D device  T structure
                         Monitor                ----------- SSUM -----------
Cmd  System    Status    Timestamp  INTERVAL  Action      TIME      WEIGHT
---  --------  --------  ---------  --------  ----------  --------  ------
 _   KEY1      ACTIVE    09:44:34   86400     ISOLATE     50        50
 _   KEY2      ACTIVE    09:44:35   86400     ISOLATE     50        15
 _   KEY3      ACTIVE    09:44:34   86400     ISOLATE     50        15
 _   KEY4      ACTIVE    09:44:36   86400     ISOLATE     50        15






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

Figure 36. INGPLEX SYSTEM command dialog panel

The following command codes are available:

C Displays the online or offline status of one or more processors and any vector facilities, or ICRFs attached to those processors

E Displays the timer synchronization mode and ETR ports

I Displays IPL information

O Displays IOS-related configuration information

S Displays the number of megabytes of central and expanded storage assigned and available to the system

D Displays the device number of one or more in-/outbound signalling paths that XCF can use and information about in-/outbound XCF signalling paths to this system

T Displays detailed signalling path information for all coupling facility structures

- The **Sysplex** field shows the name of the sysplex.
- The **GRS Mode** field shows the GRS mode of the target system. The mode can be either STAR or RING.
- The **CMD** column allows you to specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name, and press ENTER.
- The **System** column shows the name of the system.
- The **Status** column shows the status of the system.
- The **Monitor Timestamp** column shows the last time stamp recorded for status monitoring on this system.
- The **INTERVAL** column shows the system failure detection interval in seconds. This interval is the time XCF allows to elapse without a status update before

assuming that the system failed.

The last three columns contain configuration data of the SFM policy (if applicable).

- The SSUM Action field shows the SSUM action. It can be one of the following:
  - ISOLATE
  - DEACTIVATE
  - RESET
  - PROMPT
  - N/A
- The SSUM TIME field shows the SSUM interval as specified in the current SFM policy.
- The SSUM WEIGHT field shows the SSUM weight specified in the current SFM policy. This value is used in sysplex reconfigurations after a signalling connectivity failure.

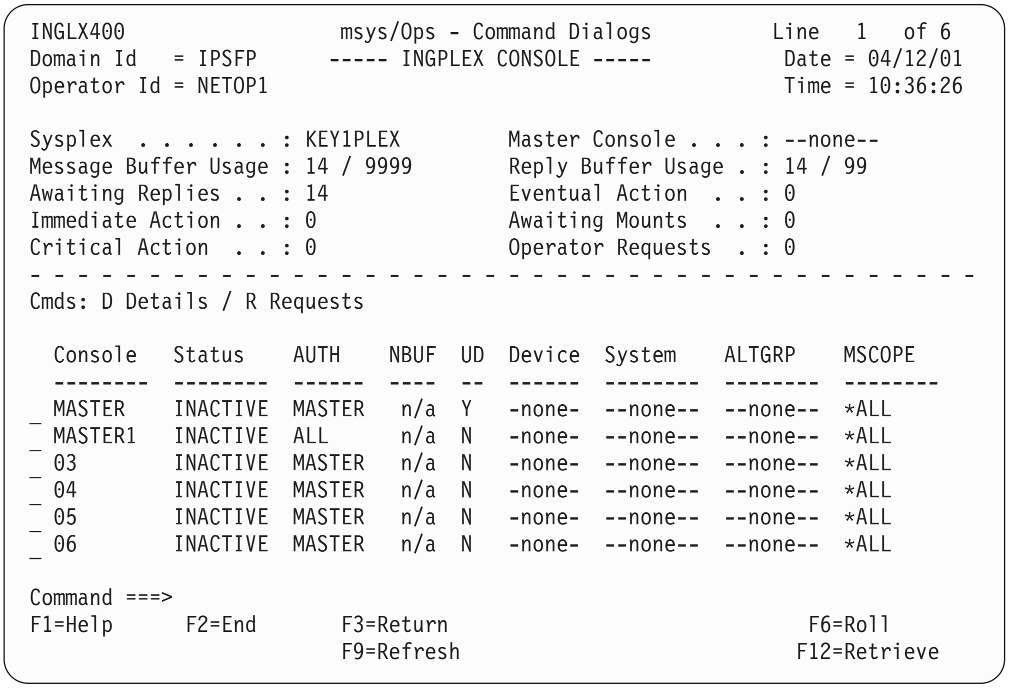
CONsole

Purpose

The CONSOLE function displays the following information for the sysplex:

- The name of the master console
- WTO & WTOR buffer utilization
- Number of queued messages (replies) of various types
- Awaiting mounts
- Operator requests and list of consoles (name, status, authority, number of WTOR buffers, UD, device, system, ALTGRP, MSCOPE)

Example



```
INGLX400                 msys/Ops - Command Dialogs         Line   1   of 6
Domain Id   = IPSFP     ----- INGPLEX CONSOLE -----          Date = 04/12/01
Operator Id = NETOP1                                         Time = 10:36:26

Sysplex  . . . . . . : KEY1PLEX        Master Console . . . : --none--
Message Buffer Usage : 14 / 9999       Reply Buffer Usage . : 14 / 99
Awaiting Replies . . : 14              Eventual Action  . . : 0
Immediate Action . . : 0               Awaiting Mounts  . . : 0
Critical Action  . . : 0               Operator Requests  . : 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: D Details / R Requests

  Console   Status    AUTH    NBUF  UD  Device  System    ALTGRP    MSCOPE
  --------  --------  ------  ----  --  ------  --------  --------  --------
_ MASTER    INACTIVE  MASTER   n/a  Y   -none-  --none--  --none--  *ALL
_ MASTER1   INACTIVE  ALL      n/a  N   -none-  --none--  --none--  *ALL
_ 03        INACTIVE  MASTER   n/a  N   -none-  --none--  --none--  *ALL
_ 04        INACTIVE  MASTER   n/a  N   -none-  --none--  --none--  *ALL
_ 05        INACTIVE  MASTER   n/a  N   -none-  --none--  --none--  *ALL
_ 06        INACTIVE  MASTER   n/a  N   -none-  --none--  --none--  *ALL
_

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

Figure 37. INGPLEX CONS command dialog panel

The following command codes are available:

D Displays details for the console

R Displays the actual requests for the console

- The **Sysplex** field shows the name of the sysplex.
- The **Message Buffer Usage** field shows the limit of the number of WTO message buffers allowed outstanding.
- The **Awaiting Replies** field shows a decimal number representing the number of messages awaiting replies.
- The **Immediate Action** field shows a decimal number representing the number of outstanding immediate action messages (with descriptor codes 1 or 2). If the number is greater than 99999, asterisks appear in this field.
- The **Critical Action** field shows a decimal number representing the number of outstanding critical eventual action messages (with descriptor code 11). If the number is greater than 99999, asterisks appear in this field.
- The **Master Console** field shows the name of the master console.
- The **Reply Buffer Usage** field shows the limit of the number of WTOR message buffers allowed outstanding. The maximum value of yyyy is specified by the RMAX parameter in the CONSOLxx parmlib member.
- The **Eventual Action** field shows a decimal number representing the number of outstanding eventual action messages (with descriptor code 3). If the number is greater than 99999, asterisks appear in this field.
- The **Awaiting Mounts** field shows a decimal number representing the number of outstanding mount requests.
- The **Operator Requests** field shows a decimal number representing the number of outstanding requests for operator intervention.
- The **CMD** column lets you specify the command codes shown on the panel. Enter the appropriate letter next to the resource name, and press ENTER.
- The **Console** column shows the name of the console as specified in the CONSOLxx parmlib member.
- The **Status** field shows the status of the console. The following values can occur:

  HARDCOPY Hardcopy log. This condition is only indicated if the console is active on the system where the command processes.

  ACTIVE Active console

  ACTIVE-P In the process of becoming an active console. This condition is only indicated if the console is active on the system where the command is processing.

  MASTER Master console

  INACTIVE Inactive console

  INACT-P In the process of becoming a non-active console. This condition is only indicated if the console is active on the system where the command is processing.

  PROB-DET The active system console is in the problem determination mode. PD is indicated only for the system console.

  SUBSYS Subsystem-allocatable console
- The **AUTH** column shows which commands may be entered from this console. The following values can occur:

  ALL Any INFO, SYS, IO, or CONS command may be entered from this console.

CONS
INFO commands and any commands from the console command group may be entered from this console.

INFO
Any command from the informational command group may be entered from this console.

IO
INFO commands and any commands from the I/O Control command group may be entered from this console.

MASTER
The specified console is authorized to enter any operator command.

NONE
This console has no command authority.

SYS
INFO commands and any commands from the system control command group may be entered form this console.

- The **NBUF** column shows the number of WTO message buffers currently queued to this console. If nnnn is greater than 9999, asterisks (****) appear in this field.
- The **UD** column shows whether this console is receiving messages with the UD attribute.
- The **Device** column shows the device number of the console as specified in the CONSOL*xx* parmlib member.
- The **System** column shows the system name of the active console.
- The **ALTGRP** column shows the alternate group defined for this console.
- The **MSCOPE** column lists the name of the system or systems from which this console is receiving unsolicited messages. Note that these systems might be different from the system where this console is physically attached.

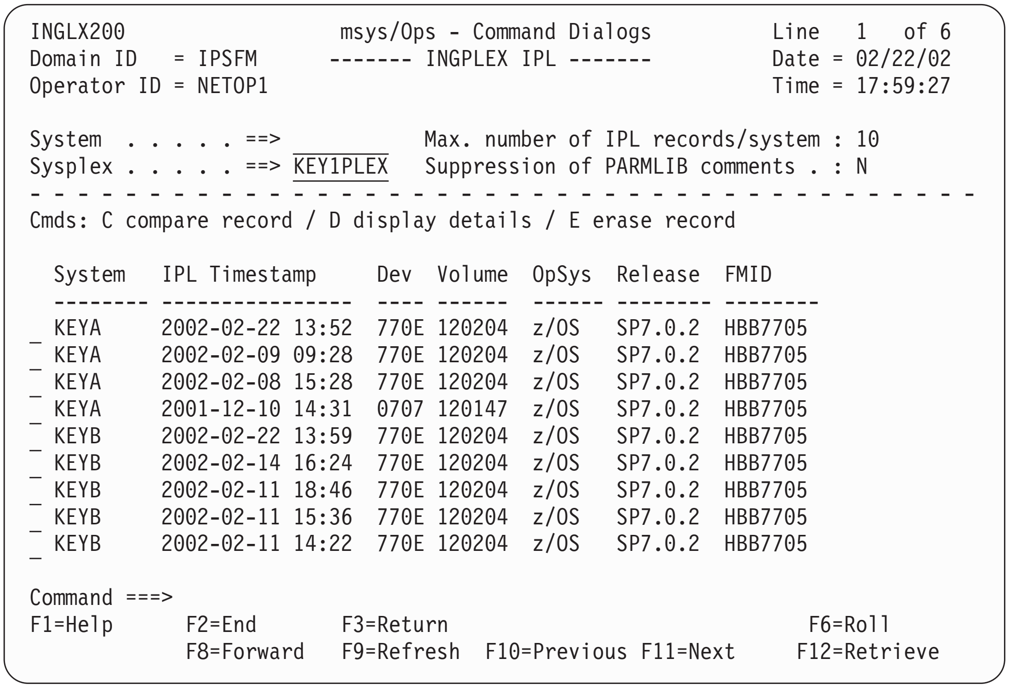
IPL

Purpose

With the INGPLEX IPL command you can view and compare the IPL information of the operating system. If a system does not behave after IPL as expected, the IPL recording function enables you to identify parameters that were changed, for example, since the last IPL. The recording function enables you to compare different IPL scenarios. INGPLEX IPL is a tool that helps to identify and resolve the cause of startup problems. The following information can be displayed:

- The selected system (or blank)
- The name of the sysplex
- The maximum number of IPLs that are stored for each system
- An indicator showing whether comments in PARMLIB members are ignored when collecting information

Example



```
INGLX200                  msys/Ops - Command Dialogs          Line   1   of 6
Domain ID   = IPSFM      ------- INGPLEX IPL -------          Date = 02/22/02
Operator ID = NETOP1                                          Time = 17:59:27

System  . . . . . ==> ________   Max. number of IPL records/system : 10
Sysplex . . . . . ==> KEY1PLEX   Suppression of PARMLIB comments . : N
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: C compare record / D display details / E erase record

  System   IPL Timestamp     Dev  Volume  OpSys  Release  FMID
  -------- ----------------  ---- ------  ------ -------- --------
_ KEYA     2002-02-22 13:52  770E 120204  z/OS   SP7.0.2  HBB7705
_ KEYA     2002-02-09 09:28  770E 120204  z/OS   SP7.0.2  HBB7705
_ KEYA     2002-02-08 15:28  770E 120204  z/OS   SP7.0.2  HBB7705
_ KEYA     2001-12-10 14:31  0707 120147  z/OS   SP7.0.2  HBB7705
_ KEYB     2002-02-22 13:59  770E 120204  z/OS   SP7.0.2  HBB7705
_ KEYB     2002-02-14 16:24  770E 120204  z/OS   SP7.0.2  HBB7705
_ KEYB     2002-02-11 18:46  770E 120204  z/OS   SP7.0.2  HBB7705
_ KEYB     2002-02-11 15:36  770E 120204  z/OS   SP7.0.2  HBB7705
_ KEYB     2002-02-11 14:22  770E 120204  z/OS   SP7.0.2  HBB7705

Command ===>
F1=Help      F2=End       F3=Return                              F6=Roll
             F8=Forward   F9=Refresh  F10=Previous F11=Next     F12=Retrieve
```

Figure 38. INGPLEX IPL main panel

Use F10 and F11 to scroll through all available columns. SORT by column numbers is supported as well as the FIND and RFind command to locate information on the panel. You can also limit the display to a particular system by specifying the system name in the appropriate entry field.

The following command codes are available:

C Compares the complete IPL information with another IPL record. A second panel will be displayed where you can select the second record.

D Displays detailed information about this IPL record.

E Erases the IPL information records. This action must be confirmed.

- The **Sysplex** field shows the name of the sysplex.
- The **System** field shows the name of the system in the sysplex.
- The **IPL Timestamp** field shows the date and time of the IPL. The format is YYYY-MM-DD HH:MM converted to local time zone.
- The **Dev** field shows the IPL device number.
- The **Volume** field shows the volume serial of the IPL device.
- The **OpSys** field shows the name of the operating system, for example, z/OS or OS/390.
- The **Release** field shows the release level of the operating system.
- The **FMID** field shows the FMID of the operating system.

For further information about the panel fields refer to the online help.

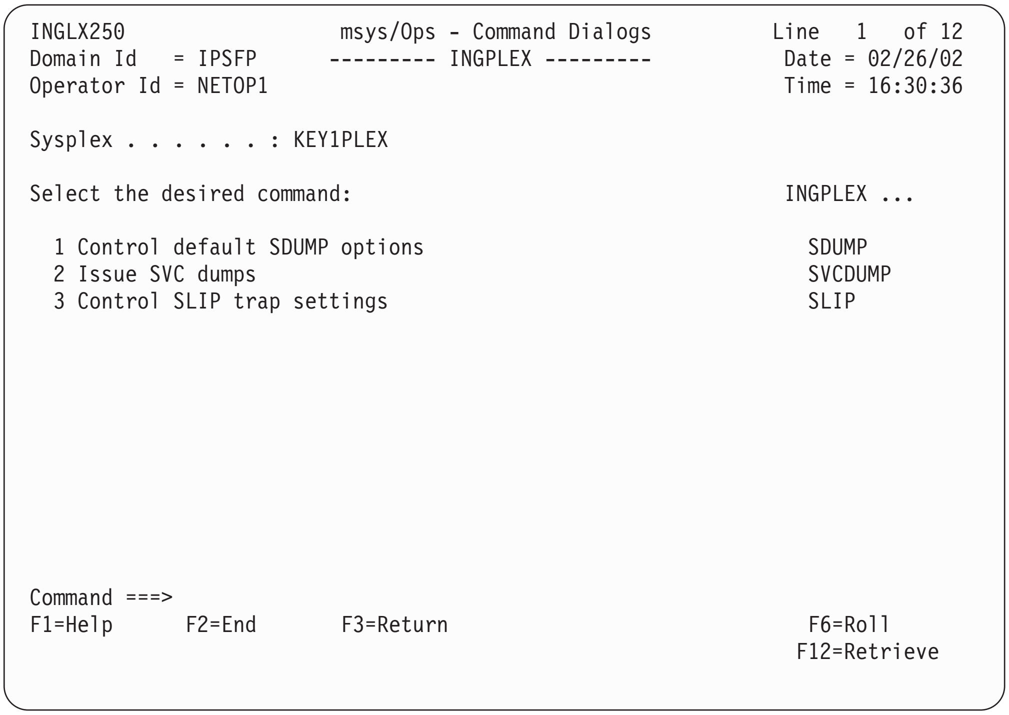
SDUMP

Purpose

The INGPLEX SDUMP command lets you control the default dump options sysplex-wide.

Example

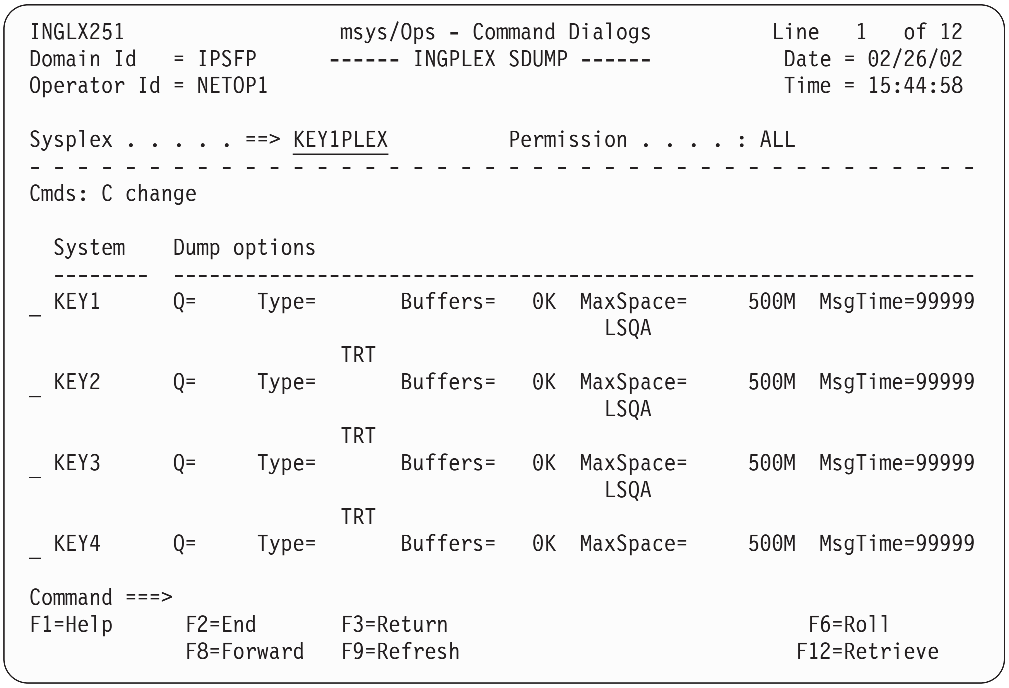
The dump functions can be invoked directly by specifying the commands, or from the dump panel of the INGPLEX command selecting the appropriate command. In addition, you can invoke the dump submenu from the main panel of the INGPLEX command selecting command **7**. The following panel is displayed:



```
INGLX250                 msys/Ops - Command Dialogs          Line   1   of 12
Domain Id   = IPSFP      --------- INGPLEX ---------          Date = 02/26/02
Operator Id = NETOP1                                          Time = 16:30:36

Sysplex . . . . . . : KEY1PLEX

Select the desired command:                                   INGPLEX ...

  1 Control default SDUMP options                               SDUMP
  2 Issue SVC dumps                                             SVCDUMP
  3 Control SLIP trap settings                                  SLIP









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

Figure 39. INGPLEX dump options panel

If you select option 1, the following panel is displayed:



```
INGLX251                  msys/Ops - Command Dialogs          Line   1   of 12
Domain Id   = IPSFP      ------ INGPLEX SDUMP ------           Date = 02/26/02
Operator Id = NETOP1                                           Time = 15:44:58

Sysplex . . . . . ==> KEY1PLEX          Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: C change

  System    Dump options
  --------  --------------------------------------------------------------------
_ KEY1      Q=     Type=       Buffers=   0K  MaxSpace=     500M  MsgTime=99999
                                                LSQA
                          TRT
_ KEY2      Q=     Type=       Buffers=   0K  MaxSpace=     500M  MsgTime=99999
                                                LSQA
                          TRT
_ KEY3      Q=     Type=       Buffers=   0K  MaxSpace=     500M  MsgTime=99999
                                                LSQA
                          TRT
_ KEY4      Q=     Type=       Buffers=   0K  MaxSpace=     500M  MsgTime=99999

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

Figure 40. INGPLEX SDUMP panel

The following command code is available:

C change
Invokes the modification panel by providing the options of the selected system as input

- The Sysplex field shows the name of the sysplex.
- The System field shows the name of the system in the sysplex.
- The Permission field shows your authorization level.
- The Dump options field shows the default SDUMP options of all systems in the sysplex. For each system the following details are displayed:

  Q= Shows whether or not SDUMP quiesces the system while dumping the contents of the SQA or CSA.

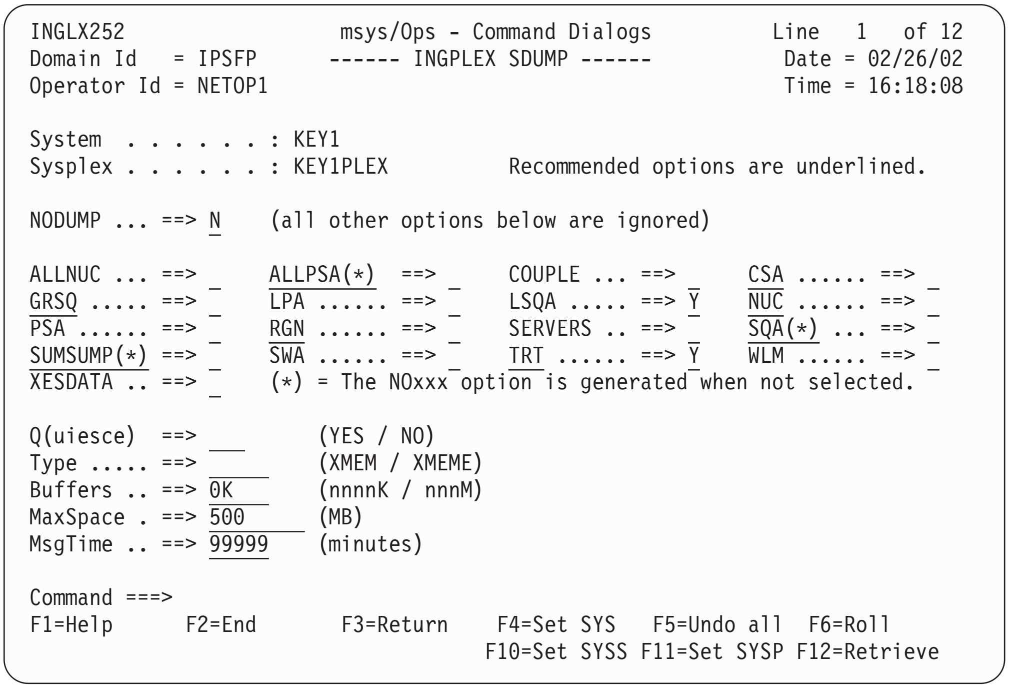
  TYPE=
  Causes SVC dump to dump the cross memory address spaces that the caller has when SVC dump gets control (XMEM) or when the error causing the dump occurs (XMEME).

  BUFFERS=
  Shows the reserved storage exlusively used by SVC dump. This storage can be used while capturing the contents of the common area storage.

  MaxSpace
  Shows the maximum amount of virtual storage that SVC dump can use to capture volatile virtual storage data, summary dump data, and component-specific data before writing the dump to DASD.

  MsgTime
  Shows for which amout of time (mm) the message IEA793A is shown at the console. When the system deletes the message, it also deletes the captured dump.

The FIND and RFIND commands are supported. If you specify command code C, the following panel is displayed:



```
INGLX252                  msys/Ops - Command Dialogs          Line   1   of 12
Domain Id   = IPSFP      ------ INGPLEX SDUMP ------           Date = 02/26/02
Operator Id = NETOP1                                           Time = 16:18:08

System  . . . . . . . : KEY1
Sysplex . . . . . . . : KEY1PLEX          Recommended options are underlined.

NODUMP ... ==> N    (all other options below are ignored)

ALLNUC ... ==> _    ALLPSA(*)  ==> _    COUPLE ... ==> _    CSA ...... ==> _
GRSQ ..... ==> _    LPA ...... ==> _    LSQA ..... ==> Y    NUC ...... ==> _
PSA ...... ==> _    RGN ...... ==> _    SERVERS .. ==> _    SQA(*) ... ==> _
SUMSUMP(*) ==> _    SWA ...... ==> _    TRT ...... ==> Y    WLM ...... ==> _
XESDATA .. ==> _    (*) = The NOxxx option is generated when not selected.

Q(uiesce)  ==> ___      (YES / NO)
Type ..... ==> _____    (XMEM / XMEME)
Buffers .. ==> 0K       (nnnnK / nnnM)
MaxSpace . ==> 500      (MB)
MsgTime .. ==> 99999    (minutes)

Command ===>
F1=Help      F2=End       F3=Return    F4=Set SYS   F5=Undo all  F6=Roll
                                      F10=Set SYSS F11=Set SYSP F12=Retrieve
```

Figure 41. INGPLEX SDUMP modification panel

The modification panel allows you to modify all SDUMP options. Furthermore, you can delete SDUMP options. After entering your changes you can set the new options for:

- The selected system
- All systems in the sysplex
- Selected systems in the sysplex

To set the options press the appropriate F-key. If you want to modify selected systems in the sysplex, you are prompted for the systems on which the SDUMP options are being changed. To reset the options to the state when the modification panel was invoked press F5 Undo all.

Note: The user must be authorized to change any SDUMP option. The authorization can be any of those which are used for controlling coupling facilities and couple data sets.

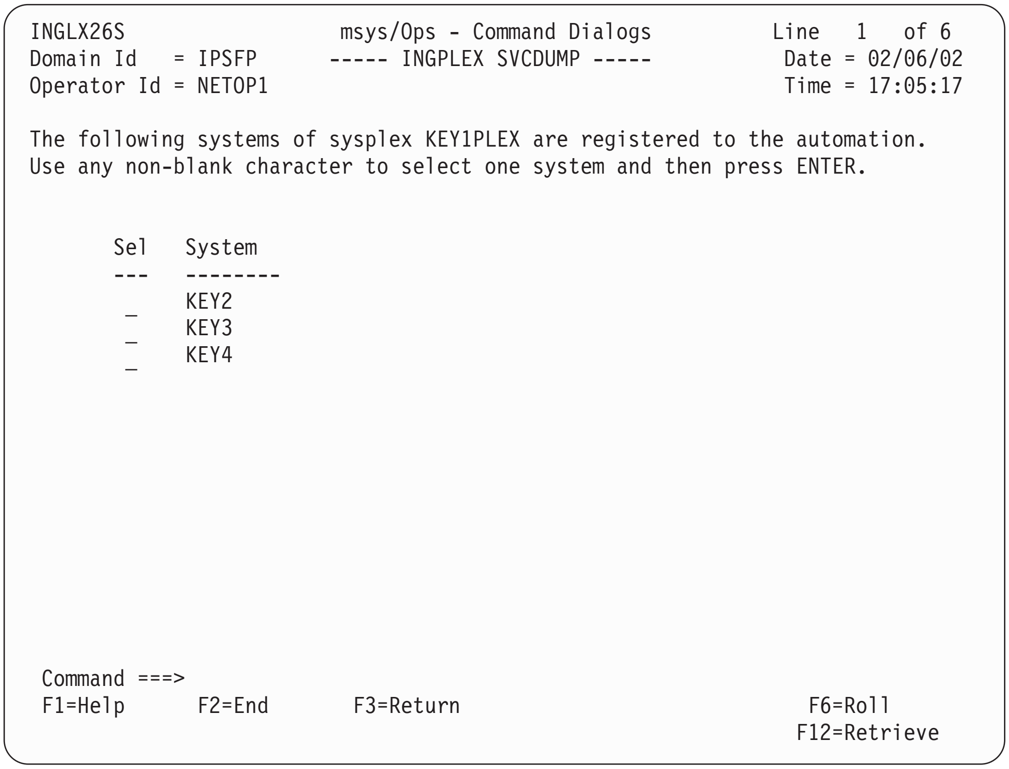
For further information about the panel fields refer to the online help.

SVCdump

Purpose

The INGPLEX SVCDUMP function allows you to issue a multisystem dump of up to 15 address spaces of a single system including their data spaces and structures.

Example

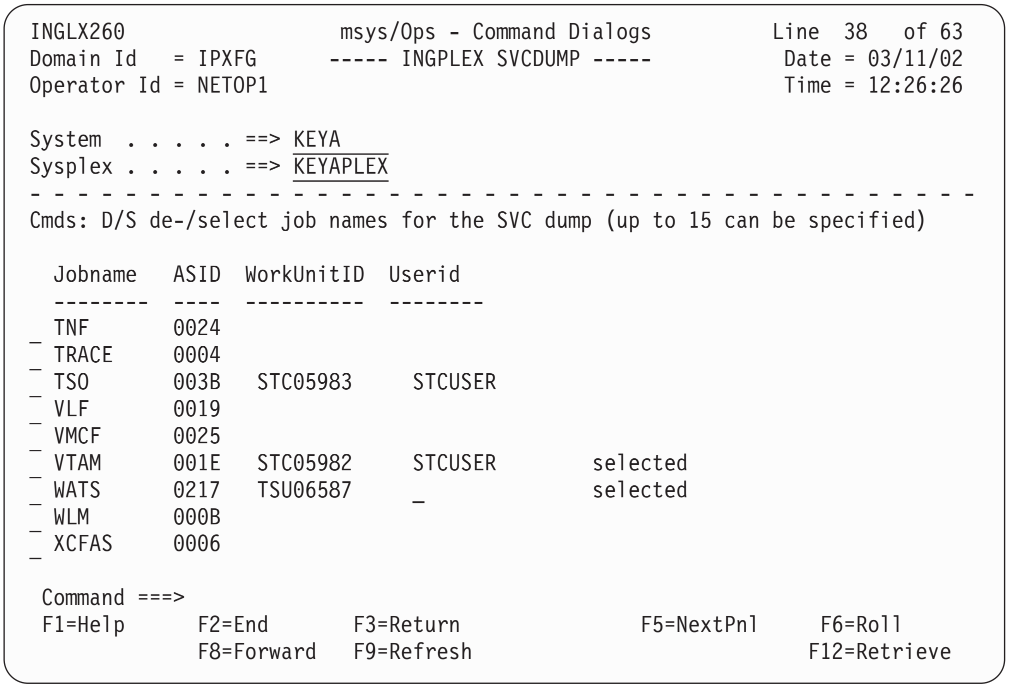


```
INGLX26S                 msys/Ops - Command Dialogs          Line   1   of 6
Domain Id   = IPSFP      ----- INGPLEX SVCDUMP -----          Date = 02/06/02
Operator Id = NETOP1                                          Time = 17:05:17

The following systems of sysplex KEY1PLEX are registered to the automation.
Use any non-blank character to select one system and then press ENTER.


       Sel   System
       ---   --------
        _    KEY2
        _    KEY3
        _    KEY4











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

Figure 42. INGPLEX SVCDUMP target system selection panel

- The Sel field lets you select a system from which a multisystem dump is issued.
- The System field shows the name of the system having joined the same XCF group the operator is logged on to.

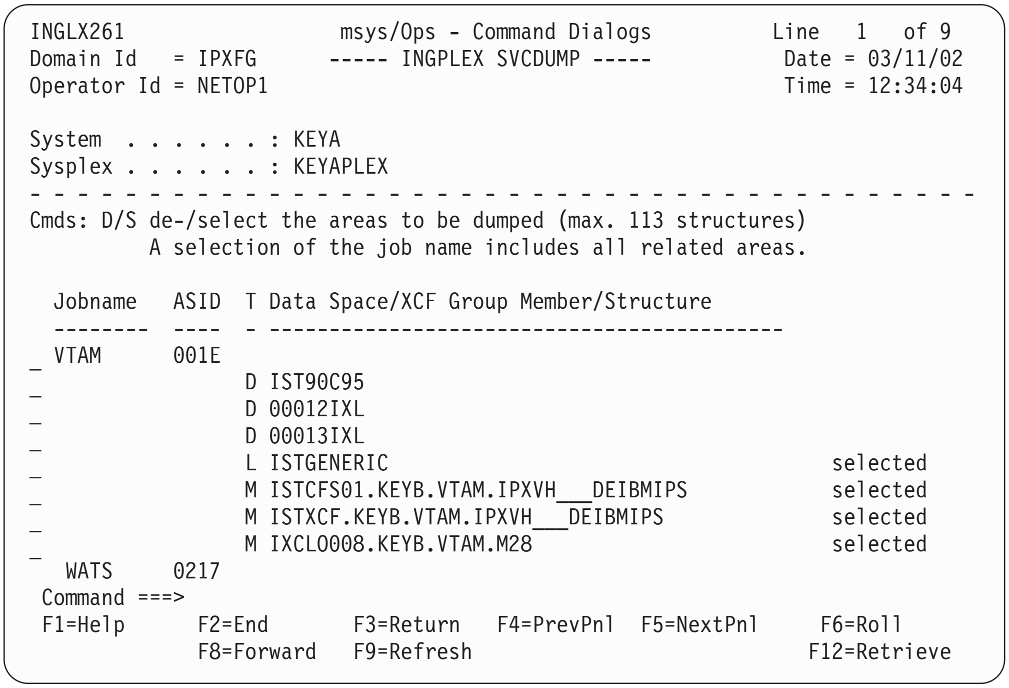
For further information about the panel fields refer to the online help. After selecting a system and pressing ENTER, the following panel is displayed:



```
INGLX260                 msys/Ops - Command Dialogs         Line  38   of 63
Domain Id   = IPXFG      ----- INGPLEX SVCDUMP -----         Date = 03/11/02
Operator Id = NETOP1                                         Time = 12:26:26

System  . . . . . ==> KEYA
Sysplex . . . . . ==> KEYAPLEX
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: D/S de-/select job names for the SVC dump (up to 15 can be specified)

  Jobname   ASID  WorkUnitID  Userid
  --------  ----  ----------  --------
_ TNF       0024
_ TRACE     0004
_ TSO       003B   STC05983     STCUSER
_ VLF       0019
_ VMCF      0025
_ VTAM      001E   STC05982     STCUSER        selected
_ WATS      0217   TSU06587     _              selected
_ WLM       000B
_ XCFAS     0006

 Command ===>
 F1=Help      F2=End       F3=Return              F5=NextPnl     F6=Roll
              F8=Forward   F9=Refresh                           F12=Retrieve
```

Figure 43. INGPLEX SVCDUMP address space selection panel

If you select VTAM address space and WATS address space, which is a user, press ENTER, then F5, the following panel is displayed:



```
INGLX261                 msys/Ops - Command Dialogs         Line   1   of 9
Domain Id   = IPXFG      ----- INGPLEX SVCDUMP -----         Date = 03/11/02
Operator Id = NETOP1                                         Time = 12:34:04

System  . . . . . . : KEYA
Sysplex . . . . . . : KEYAPLEX
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: D/S de-/select the areas to be dumped (max. 113 structures)
          A selection of the job name includes all related areas.

  Jobname   ASID  T Data Space/XCF Group Member/Structure
  --------  ----  - -------------------------------------------
_ VTAM      001E
_                 D IST90C95
_                 D 00012IXL
_                 D 00013IXL
_                 L ISTGENERIC                                       selected
_                 M ISTCFS01.KEYB.VTAM.IPXVH___DEIBMIPS              selected
_                 M ISTXCF.KEYB.VTAM.IPXVH___DEIBMIPS                selected
_                 M IXCLO008.KEYB.VTAM.M28                           selected
   WATS     0217
 Command ===>
 F1=Help      F2=End       F3=Return   F4=PrevPnl  F5=NextPnl     F6=Roll
              F8=Forward   F9=Refresh                           F12=Retrieve
```

Figure 44. INGPLEX SVCDUMP address space detail panel

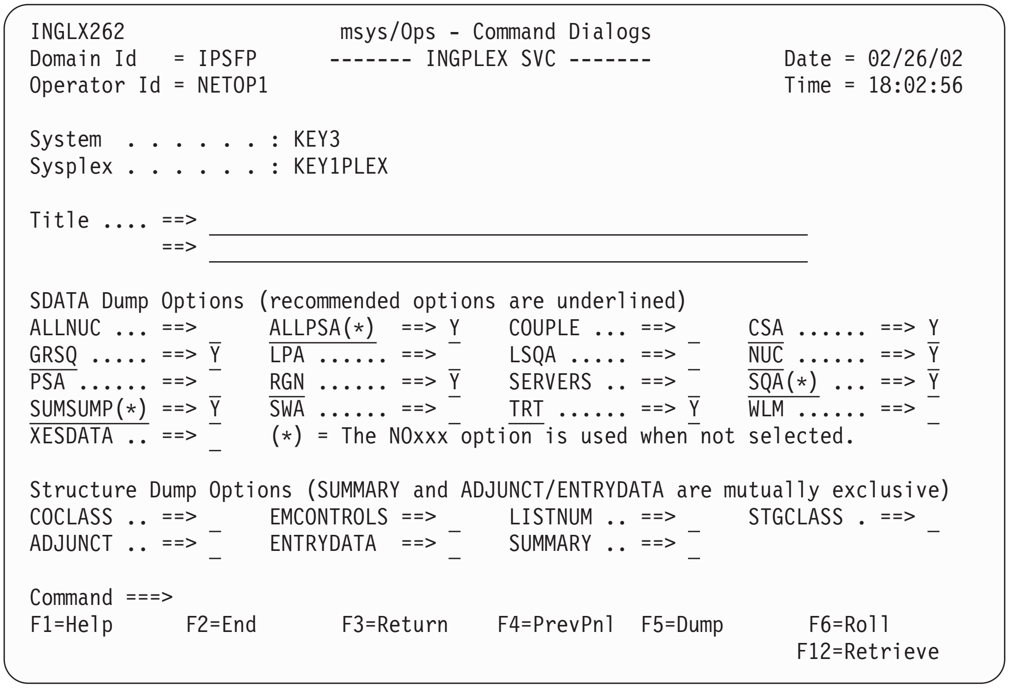
Address space VTAM has some data spaces (D), one list structure (L) and some XCF group members (M). TSO user WATS has nothing.

The following command codes are supported:

D Deselects the previous selection

S Selects a local address space, data space, structure, or XCF group member address space for the SVC dump.

If you press F5, the dump option selection panel is displayed:



```
INGLX262                  msys/Ops - Command Dialogs
Domain Id   = IPSFP      ------- INGPLEX SVC -------          Date = 02/26/02
Operator Id = NETOP1                                          Time = 18:02:56

System  . . . . . . : KEY3
Sysplex . . . . . . : KEY1PLEX

Title .... ==> __________________________________________________
           ==> __________________________________________________

SDATA Dump Options (recommended options are underlined)
ALLNUC ... ==> _     ALLPSA(*)  ==> Y    COUPLE ... ==> _     CSA ...... ==> Y
GRSQ ..... ==> Y     LPA ...... ==> _    LSQA ..... ==> _     NUC ...... ==> Y
PSA ...... ==> _     RGN ...... ==> Y    SERVERS .. ==> _     SQA(*) ... ==> Y
SUMSUMP(*) ==> Y     SWA ...... ==> _    TRT ...... ==> Y     WLM ...... ==> _
XESDATA .. ==> _     (*) = The NOxxx option is used when not selected.

Structure Dump Options (SUMMARY and ADJUNCT/ENTRYDATA are mutually exclusive)
COCLASS .. ==> _     EMCONTROLS ==> _    LISTNUM .. ==> _     STGCLASS . ==> _
ADJUNCT .. ==> _     ENTRYDATA  ==> _    SUMMARY .. ==> _

Command ===>
F1=Help      F2=End       F3=Return    F4=PrevPnl  F5=Dump       F6=Roll
                                                                F12=Retrieve
```

Figure 45. INGPLEX SVCDUMP dump option panel

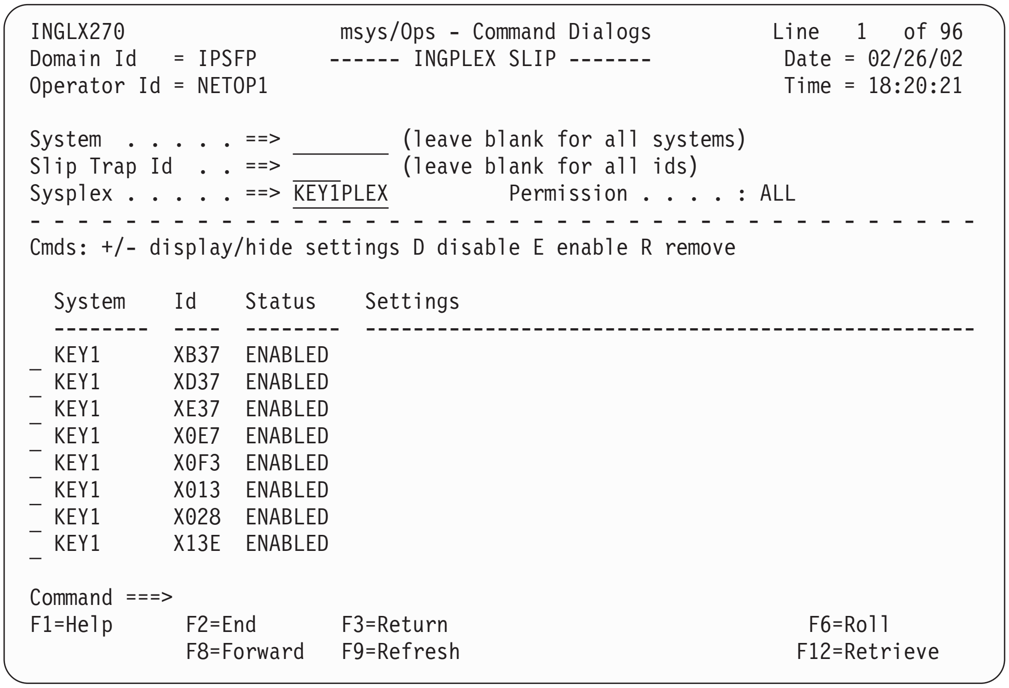
The panel shows the default dump options being set on invocation. After specifying the dump title, press F5 to issue the dump. When the dump is taken, the function returns to the address space selection panel with all selections cleared. The SORT, FIND and RFIND commands are supported for selection panels only. For further information about the panel fields refer to the online help.

SLIP

Purpose

With the INGPLEX SLIP command you can display serviceability level indication processing (SLIP) traps being set at all systems in the sysplex. With INGPLEX SLIP you can view, enable, disable, and delete the SLIP trap defined in the sysplex.

Example



```
INGLX270                  msys/Ops - Command Dialogs          Line   1   of 96
Domain Id   = IPSFP      ------ INGPLEX SLIP -------           Date = 02/26/02
Operator Id = NETOP1                                           Time = 18:20:21

System  . . . . . ==> ________ (leave blank for all systems)
Slip Trap Id  . . ==> ____     (leave blank for all ids)
Sysplex . . . . . ==> KEY1PLEX          Permission . . . . : ALL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cmds: +/- display/hide settings D disable E enable R remove

  System    Id    Status    Settings
  --------  ----  --------  ---------------------------------------------------
_ KEY1      XB37  ENABLED
_ KEY1      XD37  ENABLED
_ KEY1      XE37  ENABLED
_ KEY1      X0E7  ENABLED
_ KEY1      X0F3  ENABLED
_ KEY1      X013  ENABLED
_ KEY1      X028  ENABLED
_ KEY1      X13E  ENABLED

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

Figure 46. INGPLEX SLIP main panel

The following command codes are available:

+ Shows the settings of the SLIP trap.

– Hides the settings of the SLIP trap.

D Disables the SLIP trap.

E Enables the SLIP trap.

R Deletes the SLIP trap.

The SORT, FIND and RFIND commands are supported.

Note: The user must be authorized to enable, disable, and delete a SLIP trap. The authorization can be any of those which are used for controlling coupling facilities and couple data sets.

For information about the panel fields refer to the online help.

Chapter 15. Debugging and support commands

ACF

Purpose

The ACF command is used to load automation control file data. You can use ACF to refresh data of a particular system.

The ACF COLD command is used to activate configuration changes you have made in the msys for Operations configuration file AOFCUST.

The following list shows the most important messages issued during the operation of the ACF COLD command:

```
U IPUFD ING900I CUSTOMIZATION MEMBER AOFCUST HAS BEEN PROCESSED SUCCESSFULLY
U IPUFD AOF100I 09:17:52 : 'ACF COLD' COMMAND ISSUED
| IPUFD AOF782I AUTOMATION CONTROL FILE PROCESSING COMPLETED
```

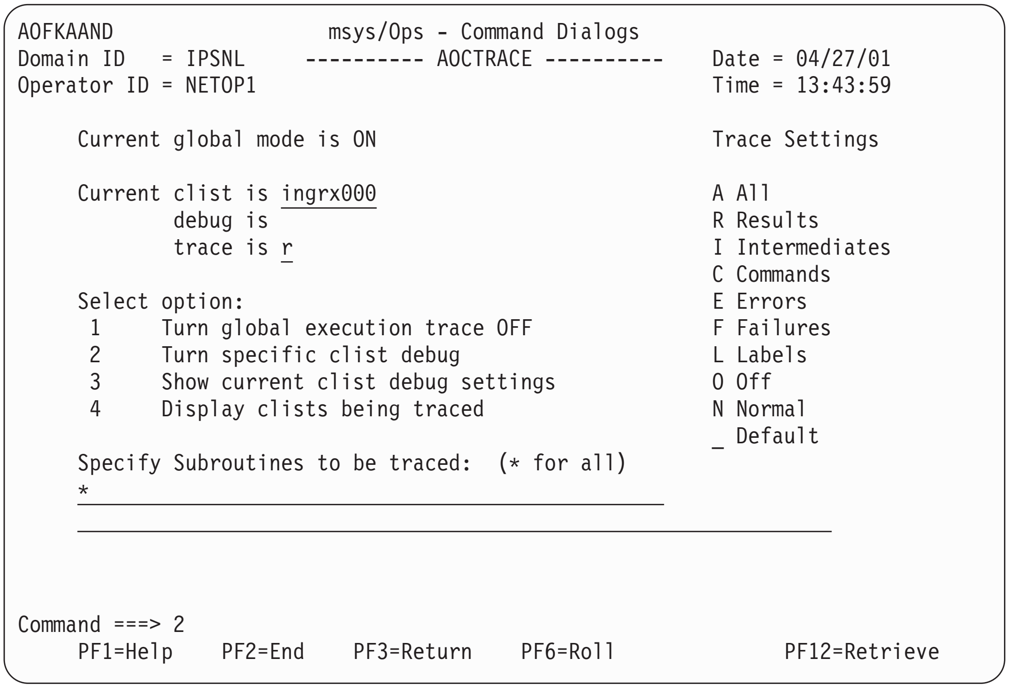
AOCTRACE

Purpose

The AOCTRACE command turns the debugging feature on or off. It works with both a global debugging feature, and two automation procedure (REXX EXEC) specific ones.

Examples

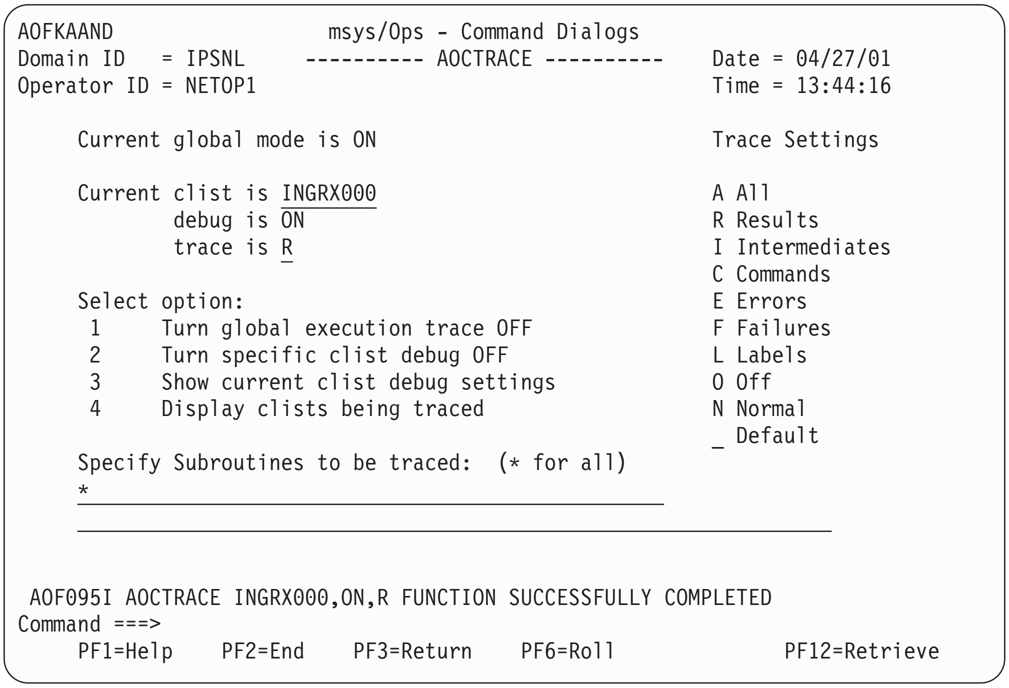
If you enter `aoctrace on` you see a message indicating that the debugging facility has been enabled. While the debugging facility is enabled, message AOF700I is written to the netlog for each procedure that is being processed. If you enter `aoctrace` a panel similar to that in Figure 47 on page 190 is displayed:



```
AOFKAAND                  msys/Ops - Command Dialogs
Domain ID   = IPSNL     ---------- AOCTRACE ----------    Date = 04/27/01
Operator ID = NETOP1                                      Time = 13:43:59

     Current global mode is ON                            Trace Settings

     Current clist is ingrx000                            A All
             debug is                                     R Results
             trace is r                                   I Intermediates
                                                          C Commands
     Select option:                                       E Errors
      1     Turn global execution trace OFF               F Failures
      2     Turn specific clist debug                     L Labels
      3     Show current clist debug settings             O Off
      4     Display clists being traced                   N Normal
                                                          _ Default
     Specify Subroutines to be traced:  (* for all)
     *




Command ===> 2
     PF1=Help    PF2=End    PF3=Return    PF6=Roll            PF12=Retrieve
```

Figure 47. AOCTRACE command dialog panel 1

If you press Enter, a panel similar to that in Figure 48 is displayed:



```
AOFKAAND                  msys/Ops - Command Dialogs
Domain ID   = IPSNL     ---------- AOCTRACE ----------    Date = 04/27/01
Operator ID = NETOP1                                      Time = 13:44:16

     Current global mode is ON                            Trace Settings

     Current clist is INGRX000                            A All
             debug is ON                                  R Results
             trace is R                                   I Intermediates
                                                          C Commands
     Select option:                                       E Errors
      1     Turn global execution trace OFF               F Failures
      2     Turn specific clist debug OFF                 L Labels
      3     Show current clist debug settings             O Off
      4     Display clists being traced                   N Normal
                                                          _ Default
     Specify Subroutines to be traced:  (* for all)
     *



 AOF095I AOCTRACE INGRX000,ON,R FUNCTION SUCCESSFULLY COMPLETED
Command ===>
     PF1=Help    PF2=End    PF3=Return    PF6=Roll            PF12=Retrieve
```

Figure 48. AOCTRACE command dialog panel 2

To switch on the trace for member `ingrx000` enter the following:

```
aoctrace ingrx000,on,r
```

To switch off the trace for member `ingrx000` enter the following:

```
aoctrace ingrx000,off,r
```

Messages

The following messages are issued (the first when the trace is switched on, and the second when it is switched off):

```
AOF302I 13:44:52 : REQUEST AOFRADBG INGRX000 ON R BY NETOP1 IS COMPLETED FOR NETOP1
AOF302I 13:45:18 : REQUEST AOFRADBG INGRX000 OFF BY NETOP1 IS COMPLETED FOR NETOP1
```

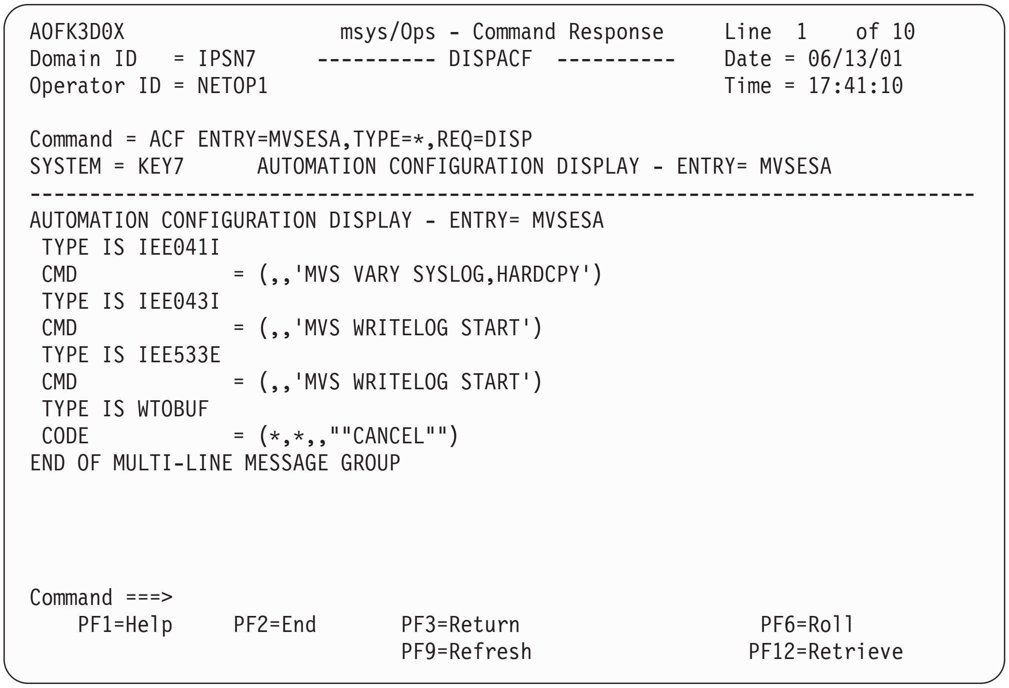
DISPACF

Purpose

The DISPACF command displays resource information and automation policy settings for a specific entry or entry-type pair in the automation control file.

Examples

If you enter `dispacf subsystem`, a panel similar to the following is displayed:



```
AOFK3D0X                  msys/Ops - Command Response     Line  1    of 10
Domain ID   = IPSN7     ---------- DISPACF  ----------    Date = 06/13/01
Operator ID = NETOP1                                      Time = 17:41:10

Command = ACF ENTRY=MVSESA,TYPE=*,REQ=DISP
SYSTEM = KEY7      AUTOMATION CONFIGURATION DISPLAY - ENTRY= MVSESA
-------------------------------------------------------------------------------
AUTOMATION CONFIGURATION DISPLAY - ENTRY= MVSESA
 TYPE IS IEE041I
 CMD             = (,,'MVS VARY SYSLOG,HARDCPY')
 TYPE IS IEE043I
 CMD             = (,,'MVS WRITELOG START')
 TYPE IS IEE533E
 CMD             = (,,'MVS WRITELOG START')
 TYPE IS WTOBUF
 CODE            = (*,*,,""CANCEL"")
END OF MULTI-LINE MESSAGE GROUP




Command ===>
    PF1=Help     PF2=End       PF3=Return                    PF6=Roll
                               PF9=Refresh                  PF12=Retrieve
```

Figure 49. Display of Automation Control File settings for subsystem (DISPACF SUBSYSTEM)

This command displays information for all types of the MVSESA entry, because you accepted the default TYPE=*.

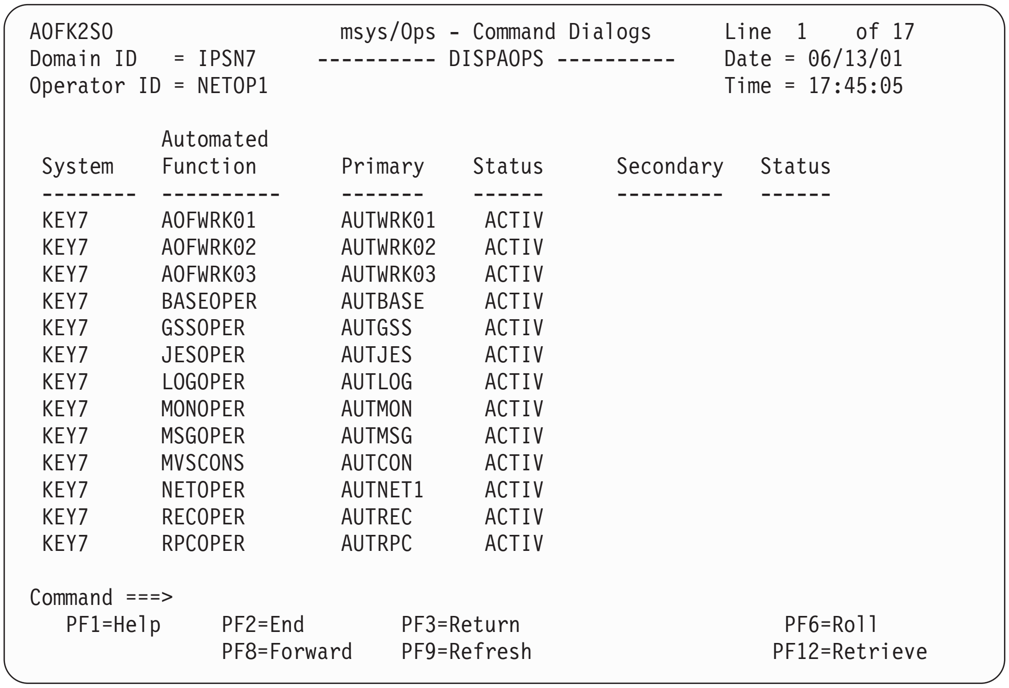
DISPAOPS

Purpose

The DISPAOPS command lists the automation operators that are currently active.

Examples

Enter dispaops on any command line and press the Enter key. You see a panel similar to the following:



```
AOFK2SO                  msys/Ops - Command Dialogs      Line  1    of 17
Domain ID   = IPSN7     ---------- DISPAOPS ----------    Date = 06/13/01
Operator ID = NETOP1                                      Time = 17:45:05

           Automated
 System    Function       Primary    Status      Secondary   Status
 --------  ----------     -------    ------      ---------   ------
 KEY7      AOFWRK01       AUTWRK01    ACTIV
 KEY7      AOFWRK02       AUTWRK02    ACTIV
 KEY7      AOFWRK03       AUTWRK03    ACTIV
 KEY7      BASEOPER       AUTBASE     ACTIV
 KEY7      GSSOPER        AUTGSS      ACTIV
 KEY7      JESOPER        AUTJES      ACTIV
 KEY7      LOGOPER        AUTLOG      ACTIV
 KEY7      MONOPER        AUTMON      ACTIV
 KEY7      MSGOPER        AUTMSG      ACTIV
 KEY7      MVSCONS        AUTCON      ACTIV
 KEY7      NETOPER        AUTNET1     ACTIV
 KEY7      RECOPER        AUTREC      ACTIV
 KEY7      RPCOPER        AUTRPC      ACTIV

Command ===>
   PF1=Help     PF2=End        PF3=Return                      PF6=Roll
                PF8=Forward    PF9=Refresh                    PF12=Retrieve
```

Figure 50. Automation operators panel

- The System field shows the name of the system where the automated function is defined
- The Automated Function field shows the name of the automated function used in msys for Operations automation procedures
- The Primary field shows the NVSS automation operator ID assigned to this automated function
- The Status field shows the current status of the primary automation operator
- The Secondary field shows the Backup NVSS automation operator ID assigned to this automated function
- The Status field shows the current status of the backup automation operator

The primary and backup NVSS automation operator IDs are assigned to the automated function in the command dialogs.

DISPASF

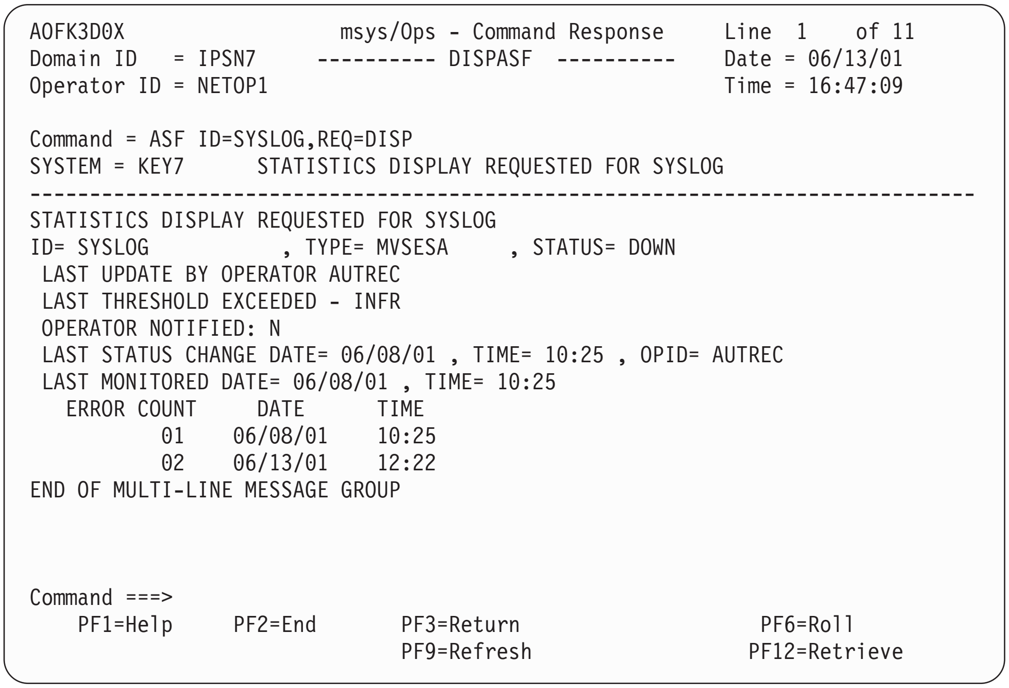
Purpose

DISPASF displays the information contained in the automation status file. This includes the:

- Automation status
- Operator ID that last changed the record
- Last threshold exceeded
- Date and time of last monitoring cycle

Examples

If you enter dispasf syslog a panel similar to the following is displayed:



```
AOFK3D0X                  msys/Ops - Command Response     Line  1    of 11
Domain ID   = IPSN7     ---------- DISPASF  ----------    Date = 06/13/01
Operator ID = NETOP1                                      Time = 16:47:09

Command = ASF ID=SYSLOG,REQ=DISP
SYSTEM = KEY7      STATISTICS DISPLAY REQUESTED FOR SYSLOG
-------------------------------------------------------------------------------
STATISTICS DISPLAY REQUESTED FOR SYSLOG
ID= SYSLOG           , TYPE= MVSESA     , STATUS= DOWN
 LAST UPDATE BY OPERATOR AUTREC
 LAST THRESHOLD EXCEEDED - INFR
 OPERATOR NOTIFIED: N
 LAST STATUS CHANGE DATE= 06/08/01 , TIME= 10:25 , OPID= AUTREC
 LAST MONITORED DATE= 06/08/01 , TIME= 10:25
   ERROR COUNT     DATE      TIME
           01    06/08/01    10:25
           02    06/13/01    12:22
END OF MULTI-LINE MESSAGE GROUP



Command ===>
    PF1=Help     PF2=End       PF3=Return                  PF6=Roll
                               PF9=Refresh                PF12=Retrieve
```

Figure 51. Display of automation status file information for DISPASF SYSLOG

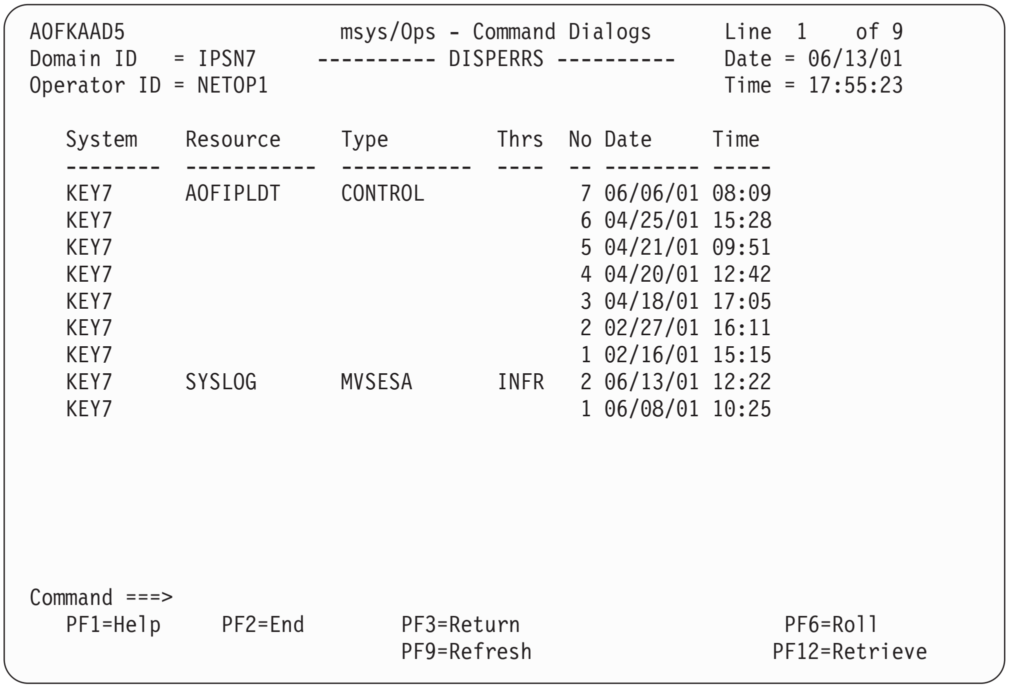
DISPERRS

Purpose

The DISPERRS command displays information about resources for which errors have been recorded in the status file.

Examples

When you enter disperrs you see a panel similar to the following:



```
AOFKAAD5                  msys/Ops - Command Dialogs      Line  1    of 9
Domain ID   = IPSN7     ---------- DISPERRS ----------    Date = 06/13/01
Operator ID = NETOP1                                      Time = 17:55:23

   System    Resource     Type         Thrs  No Date     Time
   --------  -----------  -----------  ----  -- -------- -----
   KEY7      AOFIPLDT     CONTROL             7 06/06/01 08:09
   KEY7                                       6 04/25/01 15:28
   KEY7                                       5 04/21/01 09:51
   KEY7                                       4 04/20/01 12:42
   KEY7                                       3 04/18/01 17:05
   KEY7                                       2 02/27/01 16:11
   KEY7                                       1 02/16/01 15:15
   KEY7      SYSLOG       MVSESA       INFR   2 06/13/01 12:22
   KEY7                                       1 06/08/01 10:25






Command ===>
   PF1=Help     PF2=End        PF3=Return                      PF6=Roll
                               PF9=Refresh                    PF12=Retrieve
```

Figure 52. DISPERRS command dialog panel

- The System field shows the name of the system where the resource is defined.
- The Resource field shows the name of the resource.
- The Type field shows the type of resource.
- The Thrs field shows the type of threshold that has been violated. This is either CRIT (critical), FREQ (frequent), or INFR (infrequent).
- The No field shows the sequence number assigned to the error.
- The Date and Time fields show the date and time when the error occurred.

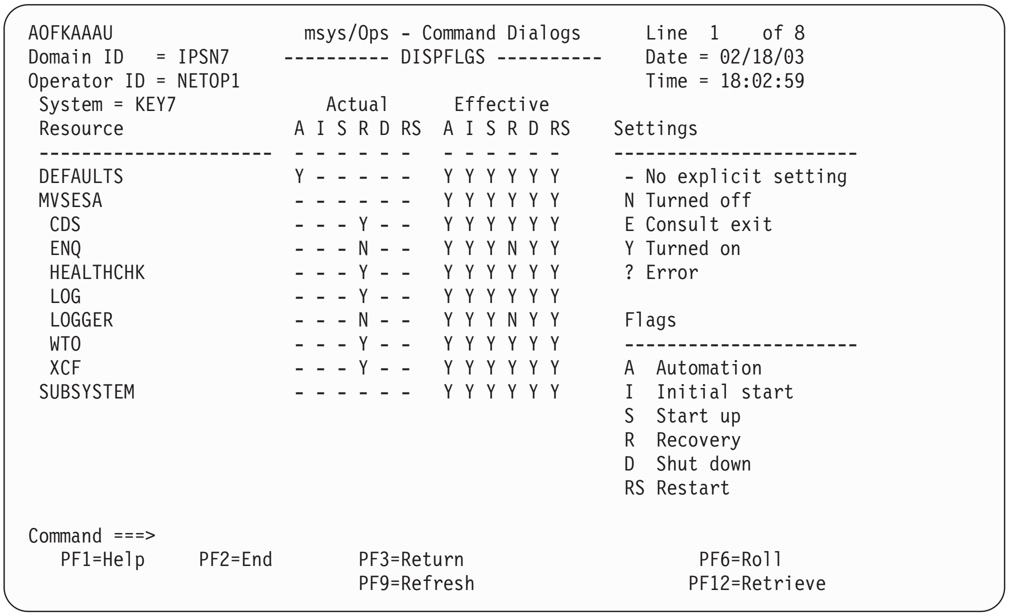
DISPFLGS

Purpose

The DISPFLGS command shows whether msys for Operations functions are enabled (Recovery flag set to Y), or disabled (Recovery flag set to N).

Examples

When you enter `dispflgs` in msys for Operations you see a panel similar to the following:



```
AOFKAAAU                 msys/Ops - Command Dialogs      Line  1    of 8
Domain ID   = IPSN7     ---------- DISPFLGS ----------   Date = 02/18/03
Operator ID = NETOP1                                     Time = 18:02:59
 System = KEY7             Actual      Effective
 Resource               A I S R D RS  A I S R D RS    Settings
 ----------------------  - - - - - -   - - - - - -     ----------------------
 DEFAULTS               Y - - - - -   Y Y Y Y Y Y      - No explicit setting
 MVSESA                 - - - - - -   Y Y Y Y Y Y      N Turned off
  CDS                   - - - Y - -   Y Y Y Y Y Y      E Consult exit
  ENQ                   - - - N - -   Y Y Y N Y Y      Y Turned on
  HEALTHCHK             - - - Y - -   Y Y Y Y Y Y      ? Error
  LOG                   - - - Y - -   Y Y Y Y Y Y
  LOGGER                - - - N - -   Y Y Y N Y Y      Flags
  WTO                   - - - Y - -   Y Y Y Y Y Y      ----------------------
  XCF                   - - - Y - -   Y Y Y Y Y Y      A  Automation
 SUBSYSTEM              - - - - - -   Y Y Y Y Y Y      I  Initial start
                                                       S  Start up
                                                       R  Recovery
                                                       D  Shut down
                                                       RS Restart

Command ===>
   PF1=Help     PF2=End        PF3=Return                  PF6=Roll
                               PF9=Refresh                PF12=Retrieve
```

Figure 53. DISPFLGS command dialog panel

As can be seen in Figure 53, the minor resource flags for CDS, HEALTHCHK, LOG, WTO, XCF are set to ON (Y) and therefore these functions have been enabled.

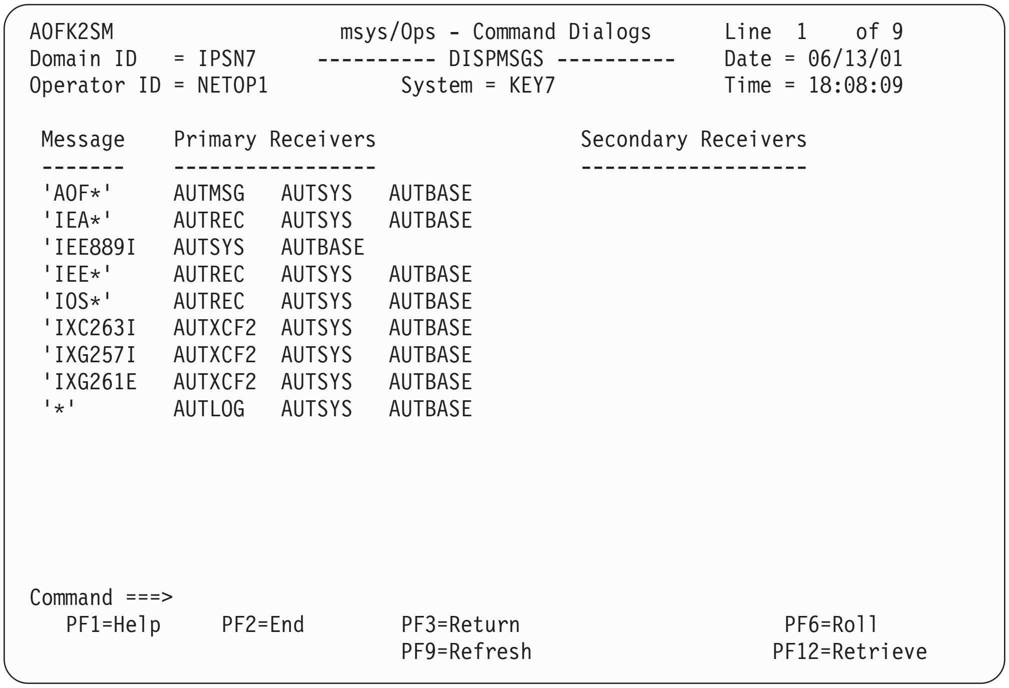
DISPMSGS

Purpose

The DISPMSGS command displays which automation operators receive each automated message.

Examples

Enter `dispmsgs` on a command line to display the Authorized Message Receivers panel:



```
AOFK2SM                  msys/Ops - Command Dialogs      Line  1    of 9
Domain ID   = IPSN7     ---------- DISPMSGS ----------    Date = 06/13/01
Operator ID = NETOP1           System = KEY7             Time = 18:08:09

 Message    Primary Receivers                Secondary Receivers
 -------    -----------------                -------------------
 'AOF*'     AUTMSG   AUTSYS   AUTBASE
 'IEA*'     AUTREC   AUTSYS   AUTBASE
 'IEE889I   AUTSYS   AUTBASE
 'IEE*'     AUTREC   AUTSYS   AUTBASE
 'IOS*'     AUTREC   AUTSYS   AUTBASE
 'IXC263I   AUTXCF2  AUTSYS   AUTBASE
 'IXG257I   AUTXCF2  AUTSYS   AUTBASE
 'IXG261E   AUTXCF2  AUTSYS   AUTBASE
 '*'        AUTLOG   AUTSYS   AUTBASE






Command ===>
   PF1=Help     PF2=End        PF3=Return                    PF6=Roll
                               PF9=Refresh                  PF12=Retrieve
```

Figure 54. Authorized Message Receivers Panel

- The Message field shows the message or message prefix.
- The Primary Receivers field shows the automation operators, identified by their NVSS IDs. The Primary Receivers column lists automation operators that can receive the messages listed beside their names. These messages go to the first automation operator listed in the Primary Receivers column that is active.
- The Secondary Receivers field shows the Alternate automation operators, identified by their NVSS IDs. Secondary Receivers receive copies of the messages listed beside their names.

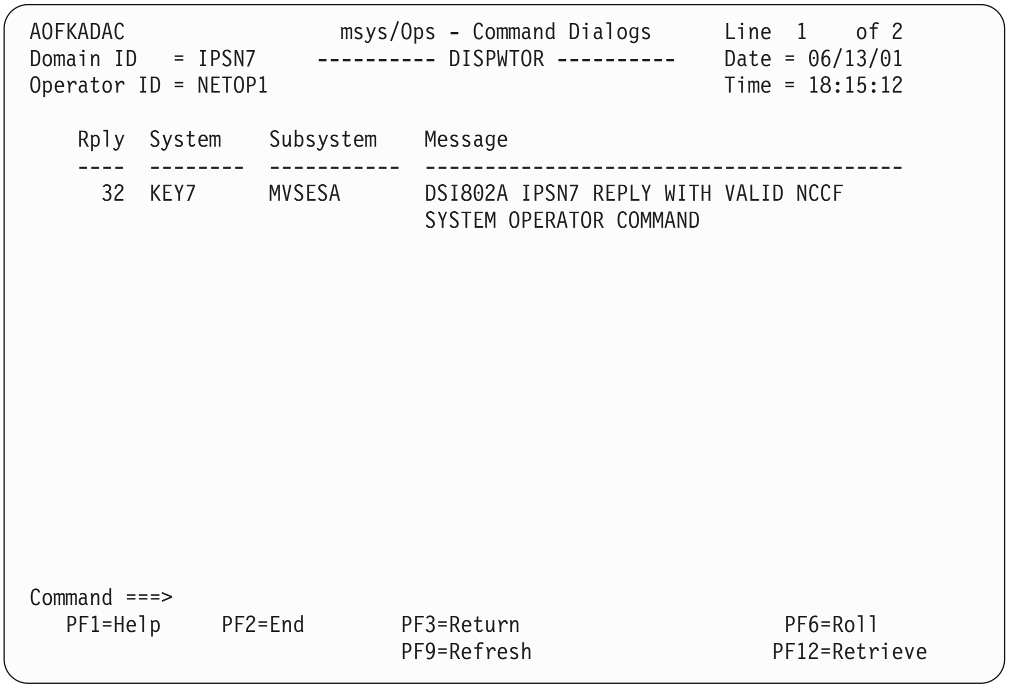
DISPWTOR

Purpose

The DISPWTOR command displays all WTORs that are currently outstanding.

Examples

If you enter `dispwtor` you see a panel similar to the following:



```
AOFKADAC                  msys/Ops - Command Dialogs      Line  1    of 2
Domain ID   = IPSN7     ---------- DISPWTOR ----------    Date = 06/13/01
Operator ID = NETOP1                                      Time = 18:15:12

    Rply  System    Subsystem    Message
    ----  --------  -----------  ----------------------------------------
      32  KEY7      MVSESA       DSI802A IPSN7 REPLY WITH VALID NCCF
                                 SYSTEM OPERATOR COMMAND

Command ===>
   PF1=Help     PF2=End        PF3=Return                     PF6=Roll
                               PF9=Refresh                   PF12=Retrieve
```

Figure 55. Display of outstanding replies (DISPWTOR)

- The Rply field shows the outstanding reply number.
- The System field shows the name of the system for which the reply is outstanding.
- The Subsystem field shows the subsystem name, or MVSESA if the reply is from a resource that is not a subsystem.
- The Message field shows the text of the message.

INGAUTO

Purpose

INGAUTO ON,MVSESA.*function_qualifier* enables msys for Operations functions. Refer to Figure 53 on page 195 for a list of the *function_qualifier* flags. INGAUTO allows you to manipulate these flags. To display the flags, use the DISPFLGS command (see “DISPFLGS” on page 194).

INGAUTO OFF,MVSESA.*function_qualifier* disables an msys for Operations function.

Examples

To turn all automation on for the WTOR buffer shortage recovery, enter the following:

```
ingauto on,mvsesa.wto
```

INGCUST

Purpose

INGCUST is a service function that is used in msys for Operations for checking the syntactical correctness of the customization member AOFCUST.

After changing AOFCUST in msys for Operations, check the syntactical correctness of the member before starting msys for Operations. If AOFCUST is syntactically incorrect, msys for Operations cannot be started. To perform the check issue INGCUST with `INSTORE` as its first parameter. INGCUST informs you about the result with a message.

INGCUST does not require that the name of the input member is AOFCUST. But in order to use a member with a different name for the customization of msys for Operations, you must rename it back to AOFCUST.

Examples

```
INGCUST INSTORE MYCUST
```

This command checks the customization member MYCUST for syntactical correctness. Note that in order to use it, you must rename it to AOFCUST.

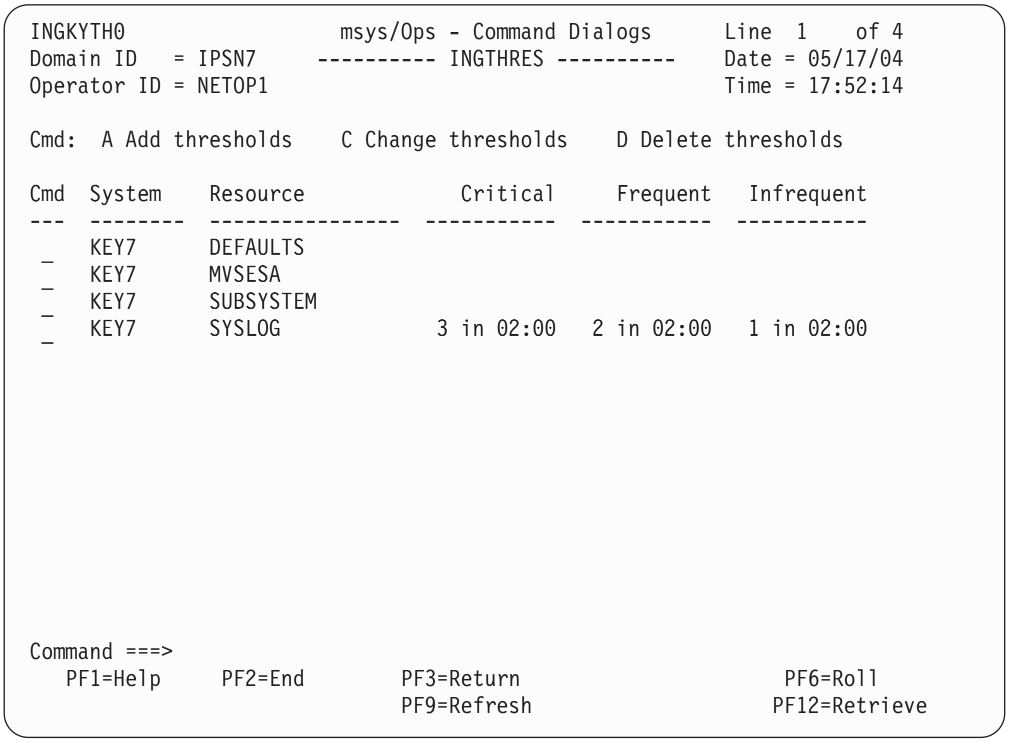
INGTHRES

Purpose

The INGTHRES command displays a summary of the msys for Operations thresholds.

Examples

When you enter `ingthres` you see a panel similar to Figure 56.



```
INGKYTH0                 msys/Ops - Command Dialogs      Line  1    of 4
Domain ID   = IPSN7     ---------- INGTHRES ----------    Date = 05/17/04
Operator ID = NETOP1                                      Time = 17:52:14

Cmd:  A Add thresholds    C Change thresholds    D Delete thresholds

Cmd  System    Resource           Critical     Frequent   Infrequent
---  --------  ----------------  -----------  -----------  -----------
 _   KEY7      DEFAULTS
 _   KEY7      MVSESA
 _   KEY7      SUBSYSTEM
 _   KEY7      SYSLOG            3 in 02:00   2 in 02:00   1 in 02:00















Command ===>
   PF1=Help     PF2=End        PF3=Return                      PF6=Roll
                               PF9=Refresh                    PF12=Retrieve
```

Figure 56. INGTHRES command dialog panel

- The System field shows the name of the system where the resource is defined.
- The Resource field shows the name of the resource. The first three entries are always DEFAULTS, MVSESA, and SUBSYSTEM.

- The Critical field shows the critical threshold for the resource. It contains an entry in the format `nn in hh:mm` which means that the threshold will be exceeded if more than *nn* errors occur for the resource within *hh* hours and *mm* minutes.

  If no critical threshold has been specified, the field is blank.
- The Frequent field shows the frequent thresholds for the resource.
- The Infrequent field shows the infrequent thresholds for the resource.

  Thresholds have the format *nn* in *hh:mm*, which means that the threshold is exceeded if more than *nn* errors occur within *hh* hours and *mm* minutes.

You can use the following command codes:

A This allows you to add new thresholds for a resource using the same threshold settings as the selected resource. Another panel will be displayed where you can specify the resource name and optionally overtype the threshold settings.

C This allows you to modify the thresholds of the selected resource. Another panel will be displayed that shows the current threshold settings. Here you can overtype the appropriate values.

D This deletes the threshold settings for the selected resource.

Part 5. Setup reference

This part provides a reference for setting up msys for Operations, and has the following chapter:

- Chapter 16, “msys for Operations definition statements reference,” on page 203

Chapter 16. msys for Operations definition statements reference

Definition statements are used by msys for Operations for performing system administration tasks. System administration is the process of redefining system defaults and storage requirements. You can perform administration subtasks during msys for Operations installation or when you redefine network resources.

In planning for installation and network management tasks, you determine the facilities you need to run msys for Operations. You also need to identify hardware requirements and specific resources used by msys for Operations. This information can help you determine requirements for user coding and msys for Operations definitions.

After copying current definition statements from the sample files, you can begin to modify existing definition statements or create new ones. You can alter definition statements during a first-time msys for Operations installation, or later, while running your production system.

Statement formats

The format of a definition statement is:

- Statement name
- General introduction

  The general introduction explains overall options, assumptions, and the purpose of the statement. Each introduction explains the name of the member and where you code the statement.
- Definition statement syntax

  The definition statement syntax is a model statement that is formatted according to the code conventions.
- Operand descriptions

  This section describes each operand you can specify for the definition statement. The description includes the specific values or variable information that you can specify for the operand.

Syntax conventions for definition statements

These syntax conventions apply to most statements:

- Code at least one blank between a label name and the name of the definition statement, and between the name of the definition statement and the first operand. One or more blanks, or a single comma with no blanks, must separate the statement operands. You cannot separate the operands with a combination of commas and blanks. If you omit the optional label name, you still need to precede the definition statement with one or more blanks.
- The label field must not exceed 8 characters, and the field must start in column 1.
- Continuation from one line to the next is not allowed. However, you can repeat the definition statement and add the remaining information. For the following example:

```
LOGINIT AUTOFLIP=YES
LOGINIT RESUME=YES
```

Is the same as:

```
LOGINIT AUTOFLIP=YES,RESUME=YES
```

- Place comments on a separate line for DSIPARM members. The first column of a comment line must contain an asterisk (*).
- Many definition files conclude with an END statement. This END statement has no operands and cannot begin in column 1.
- All NVSS program identifiers, which are called names, must not exceed 8 characters unless specified. The first character must be alphabetical and alphabetical characters must be in uppercase.
- Command names, command list names, and any other NVSS program identifiers must not contain commas (,), periods (.), blanks ( ), apostrophes (′), ampersands (&), asterisks (*), or equal signs (=). Commas, periods, blanks, and equal signs are used as delimiters when the definition statements are parsed. The other characters have special meanings for NVSS command lists.
- Command names and command list names must begin in column 1.
- System symbolics can be coded on any NVSS definition statement to provide unique information to the msys for Operations system. System symbolics are useful when running msys for Operations on different systems where you want to have different characteristics. This unique information (as defined by the system symbolic values) will remain on your system definitions until you change those definitions and re-IPL MVS.

AUTH

The AUTH statement defines an operator’s authority to view and control resources and specifies whether an operator is eligible to be the authorized receiver. Code this statement in a member specified by a PROFILEN statement associated with the operator. See “OPERATOR” on page 205 and “PROFILEN” on page 214 for information on how a PROFILEN statement is associated with an operator. A sample member supplied with msys for Operations is DSIPROFA.

The syntax for the AUTH statement is:

AUTH

```
           ┌─MSGRECVR=NO──────────┐
►►──AUTH ──┼──────────────────────┼───────────────────────────────►◄
           └─MSGRECVR=──┬─NO──┬───┘
                        └─YES─┘
```

Where:

MSGRECVR=NO|YES

Specifies whether operators using this profile can receive unsolicited messages that are not routed to a particular operator by the use of the NVSS ASSIGN command or by NVSS automation.

NO

Indicates that operators using the profile containing this statement do not receive unsolicited messages. NO is the default.

YES
Indicates that an operator using this profile can be the authorized message receiver.

Usage Notes:

- In NVSS, the ***authorized receiver*** is the operator authorized to receive all the unsolicited and authorized messages that are not routed to a specific operator with an ASSIGN command or a ROUTE action in an NVSS automation statement. The authorized receiver is determined by the order in which operator terminals are defined to NVSS and by the order in which authorized operators have logged on.
- When several operators are eligible to receive a particular message, NVSS uses the following priority order (from the lowest to the highest) to route the message to the proper operator:
  - The operator designated by an ASSIGN command
  - The operator or operators designated by the ROUTE action in the automation table
  - An operator

    If more than one operator is logged on, the one logged on first has priority.
  - An autotask operator

    If more than one autotask has been started, the one started first has priority. Use the ASSIGN command if an autotask is going to be the receiver of unsolicited messages.
  - The system console operator

OPERATOR

When OPERSEC is not specified as SAFDEF on either the OPTIONS statement in DSIDMN or on the REFRESH command, the OPERATOR statement identifies each operator who can log on to this NVSS program. This statement is also used to define operator identifiers that can be started as automation tasks by the AUTOTASK command. The OPERATOR statement must come before its associated PROFILEN statements. You code this statement in DSIOPF.

You can dynamically add or delete operators by adding or deleting OPERATOR statements in DSIOPF and issuing the REFRESH OPERS command.

The syntax for the OPERATOR statement is:

OPERATOR

```
►►──opid OPERATOR PASSWORD=password──┬──────────┬──────────────────────────►◄
                                     └─,NOCHECK─┘
```

Where:

opid
Indicates the 1–8 character value that identifies an operator. Valid characters for the operator identifier are letters A–Z, the numbers 0–9, or the special characters number sign (#), at sign (@), or dollar sign ($). The identifier must begin in column 1. Each operator must have a unique operator identifier. Also, code an operator statement for each operator identifier you want to use for an automation task. Do not use the names of hardcopy logs, terminals, or task

identifiers as operator identifiers. The following identifiers are reserved by the NVSS program and cannot be used as operator identifiers:

- ALL
- DPR
- DST
- HCL
- HCT
- LOG
- MNT
- NNT
- OPT
- OST
- PPT
- SYSOP
- TCT

Additionally, if the operator identifier is the same as the LU name (terminal), some command lists assume that the operator is an autotask and do not run.

PASSWORD=*password*
Indicates the 1–8 character operator password. You are required to code a password, but the password is ignored if you code OPTIONS OPERSEC=MINIMAL or OPERSEC=SAFCHECK in DSIDMN. The password is also ignored if you use this operator identifier when starting an autotask using the AUTOTASK command. For operator identifiers set up specifically for autotasks, use the password to identify the operator as such.

NOCHECK
Allows the NVSS operator to log on without NVSS verifying the password. In this case, message DWO354 is sent to the authorized receiver indicating that the operator has logged on and the password has not been verified by NVSS. The advantage of this option over OPERSEC=MINIMAL is that the operator's profile is used and any initial command specified is used. NOCHECK must be preceded by a comma.

OPTIONS

The OPTIONS statement describes the type of security settings msys for Operations uses. This checking includes:

- Authorizing a user to log on as an msys for Operations operator
- Validating the password and identity of a user logging on to msys for Operations
- Specifying where the attributes for an msys for Operations operator are defined
- Protecting commands executed in NVSS

Code the OPTIONS statement in DSIDMN. You can use the REFRESH command to dynamically change these options.

There can be more than one OPTIONS statement. If you specify the same keyword on more than one OPTIONS statement, the first occurrence is used. Subsequent specifications of a keyword result in an error message and initialization continues.

During initialization, messages BNH180I, BNH191I, and BNH193I are issued to display the operator and command security settings that will be used by msys for Operations. If error messages are displayed during initialization, message DSI813A is issued at the end of initialization to give the operator a chance to continue or to terminate initialization.

An OPTIONS statement is required for msys for Operations and a value must be specified for the OPERSEC and CMDAUTH keywords.

If OPERSEC=SAFDEF, you do not need a DSIPRF specification in your msys for Operations procedure. If you later want to issue a REFRESH command with OPERSEC specified as either NETVPW or SAFCHECK, you must first dynamically allocate DSIPRF, if it is not in your msys for Operations procedure.

If OPERSEC=SAFDEF, you do not need a DSIOPF member in DSIPARM. If you later want to issue a REFRESH command with OPERSEC specified as either NETVPW or SAFCHECK, DSIPARM must contain a DSIOPF member for the REFRESH command to complete successfully.

The syntax for the OPTIONS statement is:

OPTIONS

```
                              (1)
►►──┬─────────┬──OPTIONS──────────,OPERSEC=──┬─SAFCHECK─┬──────────────────────►
    └─label ──┘                              ├─MINIMAL──┤
                                             ├─NETVPW───┤
                                             └─SAFDEF───┘

   ┌─,CMDAUTH=SAF──────────────────────────────┐  ┌─,AUTHCHK=SOURCEID──────┐
►──┼───────────────────────────────────────────┼──┼────────────────────────┼──►◄
   │                ┌─,SAFNODEC=PASS─────────┐ │  └─,AUTHCHK=──┬─TARGETID─┤
   └─,CMDAUTH=──┬─SAF─┼────────────────────────┼─┘              └─SOURCEID─┘
                │     ├─,SAFNODEC=──┬─PASS─┬───┤ │
                │     │             └─FAIL─┘   │ │
                │     └─,BACKTBL=table_name────┘ │
                └─TABLE,TBLNAME=table_name───────┘
```

Notes:

1 For more information about the interrelationship of keywords, see Table 9 on page 211.

Where:

label
Indicates the optional label for the OPTIONS statement. This label identifies the statement for any related error messages.

OPERSEC=MINIMAL | NETVPW | SAFCHECK | SAFDEF
Defines the method used to allow users to log on to NVSS.

MINIMAL
Specifies that msys for Operations operators are defined by a list of operator identifiers in DSIOPF. There is no password validation. The logon profile is not used and logon operands specified on the logon menu are ignored.

Other keywords cannot be specified on the OPTIONS statement when MINIMAL is specified. If other keywords are specified, they are ignored, a message is issued, and initialization continues. Message DSI813A is issued at the end of initialization, indicating these errors and giving the operator a chance to continue or to terminate initialization.

You cannot use the REFRESH command to change the OPERSEC specification when MINIMAL is specified.

NETVPW

Specifies that msys for Operations operators are defined by a list of operator identifiers in DSIOPF. The identification is validated with a password associated with the identifier in DSIOPF. The profile, read from DSIPRF at logon, contains information about what the operator is allowed to do, and limits commands and resources that the operator can use.

CMDAUTH=SAF should not be specified when OPERSEC=NETVPW. If CMDAUTH=SAF is specified, no authorization checks are performed when commands are issued.

If an error with the CMDAUTH specification is detected, an error message is issued and initialization continues. Message DSI813A is issued at the end of initialization, indicating that errors have occurred and giving the operator a chance to continue or to terminate initialization.

SAFCHECK

Specifies that operator identification and password checking is performed using an SAF security product. The operator identifier must also be defined in DSIOPF, and other attributes given to the operator at logon are taken from the operator's specified profile in DSIPRF.

Security access checks that occur when an operator tries to access a data set that is protected in the DATASET class of an SAF product or an MVS system command that is protected in the OPERCMDS class of an SAF product are checked against the authority of the operator.

This specification allows CMDAUTH to be specified as TABLE or SAF.

SAFDEF

Specifies that operator identification and password checking is done using an SAF security product. Authority to log on as an msys for Operations operator is controlled through the APPL class. The operator identifier must be authorized to the resource name in the APPL class which represents msys for Operations.

The attributes given to the operator at logon are defined in theNVSS segment of the operator's user profile in the SAF product.

When OPERSEC=SAFDEF is specified, any value for CMDAUTH can be used.

CMDAUTH=SAF|TABLE

Defines the method used by msys for Operations to protect command usage. For a list of commands with keywords and values that can be protected, see "Recommended commands to protect" on page 81.

CMDAUTH cannot be specified when OPERSEC=MINIMAL. With other OPERSEC settings you can issue the REFRESH command to change the method used for command authorization.

If the CMDMDL statement for a command specifies SEC=BY, no authority checking is done for that command. Command authorization checking for automation table commands can also be bypassed by specifying AUTOSEC=BYPASS on the DEFAULTS command.

SAF Specifies that msys for Operations performs command authorization checking using an SAF security product. The commands you want to protect should be defined in the NETCMDS class in the security product. The operators that you want to give access to the commands can be permitted to use the resource names that represent commands, keywords, and values. For more information, see "Command authorization" on page 77.

If AUTHCHK is not specified when CMDAUTH=SAF, it defaults to SOURCEID during initialization. When the REFRESH command is used, there is no default for AUTHCHK.

If, during initialization, the NETCMDS class is not active or the security product is not active and BACKTBL is not specified, no authorization checking is performed when commands are issued. An error message is issued and initialization continues. Message DSI813A is issued at the end of initialization, indicating that errors have occurred and giving the operator a chance to continue initialization or to terminate it.

If, during command authorization checking, the NETCMDS class is inactivated, the security product is inactivated, or there is no resource name defined for the command being issued, msys for Operations uses the command authorization table specified by BACKTBL. If BACKTBL is not specified, authority to issue the command is determined by the value of SAFNODEC.

For immediate commands, authority checking is not performed by SAF. These commands should be protected using the command authorization table specified by BACKTBL.

CMDAUTH=SAF can only be specified when OPERSEC=SAFCHECK or OPERSEC=SAFDEF. If CMDAUTH=SAF is specified with any other value for OPERSEC, initialization continues, and no authorization is in effect. Message DSI813A is issued at the end of initialization, indicating that errors have occurred and giving the operator a chance to continue or to terminate initialization.

TABLE

Specifies that authorization checking is performed using a dynamic command authorization table specified by the TBLNAME keyword. The table can be modified and reloaded using the REFRESH command.

If AUTHCHK is not specified, it defaults to SOURCEID for CMDAUTH=TABLE during initialization. When the REFRESH command is used, there is no default for AUTHCHK.

If CMDAUTH=TABLE is specified and TBLNAME is not specified, or if the member specified by TBLNAME is not found, messages are issued and command authorization checks are not performed. Message DSI813A is issued at the end of initialization, indicating that errors have occurred and giving the operator a chance to continue initialization or to terminate it.

BACKTBL=*table_name*

Specifies the 1–8 character name of the command authorization table used when CMDAUTH=SAF and SAF checking cannot be performed for the command being issued. This can occur when:

- The command is an immediate command.
- There is no resource name defined in the NETCMDS class which protects or authorizes this command.

- The NETCMDS class is not active.
- The security product is not active.

The table name is a member name in a DSIPARM data set.

The BACKTBL and SAFNODEC options are mutually exclusive. When both are specified, the BACKTBL option is used.

table_name
Specifies the name of the table. For more information on how to build the table, see "Command authorization" on page 77.

If CMDAUTH=SAF is specified and the backup command authorization table contains syntax errors, messages are issued. Message DSI813A is issued at the end of initialization, indicating that errors have occurred and giving the operator a chance to continue initialization or to terminate it. If the reply to DSI813A is YES, the table will still be loaded with the statements that are coded correctly. If the backup table contains no valid statements, NVSS continues with no backup table.

AUTHCHK=SOURCEID | TARGETID
Specifies the user ID that is to be used when verifying command authorization and span authorization for VTAM commands. For specific information about how the SOURCEID and TARGETID are determined, "Command authorization" on page 77.

SOURCEID
Specifies to check the authority of the original issuer or the ID closest to the original issuer.

Access failure messages display the source issuer of the command. SOURCEID is the default when CMDAUTH=SAF.

TARGETID
Specifies to check the authority of the task under which the command runs.

SAFNODEC=PASS | FAIL
Defines the action taken by msys for Operations when BACKTBL is not specified and the SAF product cannot make a decision on command authority. This can occur when:

- There is no resource name defined in the NETCMDS class which protects or authorizes this command.
- The NETCMDS class is not active.
- The security product is not active.

The default is to pass the command.

The SAFNODEC and BACKTBL options are mutually exclusive. When both are specified, the BACKTBL is used.

PASS Specifies that users should be allowed to issue all commands when the SAF product cannot make a security decision.

FAIL Specifies that users should not be allowed to issue any commands when the SAF product cannot make a security decision.

Attention: Specifying PASS indicates that no command authorization checking is performed if the SAF product becomes unavailable or otherwise cannot make a decision. Specifying FAIL keeps you from issuing any commands (except when SEC=BY is specified on the CMDMDL statement or AUTOSEC=BYPASS is in effect for automation commands) if the SAF product becomes unavailable or otherwise cannot make a decision. For this reason, it is better to define a backup command authorization table than to use SAFNODEC.

Interrelationships of keywords:

Table 9 shows the relationships between OPERSEC and CMDAUTH specifications.

Table 9. Interrelationships between OPTIONS keyword values (part 1 of 2)

IF OPERSEC=	CMDAUTH	
	TABLE	SAF
MINIMAL	Not valid	Not valid
NETVPW	Is valid	Not valid
SAFCHECK	Is valid	Is valid
SAFDEF	Is valid	Is valid

Table 10 shows the relationships between CMDAUTH, TBLNAME, BACKTBL, SAFNODEC, and AUTHCHK specifications.

Table 10. Interrelationships between OPTIONS keyword values (part 2 of 2)

CMDAUTH=	TBLNAME	BACKTBL	SAFNODEC	AUTHCHK	
				TARGETID	SOURCEID
SAF	Ignored	Is valid	Is valid, defaults to PASS, and is ignored if BACKTBL is specified	Is valid	Default
TABLE	Required	Ignored	Ignored	Is valid	Default

Operator Authority:

Table 11 shows how the OPERSEC and OPSPAN keywords are used to specify how operator verification and authority checking is to be performed.

Table 11. Defining and verifying operator authority

Keyword	Related Defaults	Restrictions	Effect
OPERSEC= MINIMAL		• CMDAUTH is ignored. • Cannot use REFRESH to change OPERSEC.	• Logon profile ignored • Logon operands ignored • No password validation • Operator must be defined in DSIOPF
OPERSEC= NETVPW	CMDAUTH=TABLE must be specified.	• Not valid when CMDAUTH=SAF	• Password validated from DSIOPF • Operator must be defined in DSIOPF • Profile read from DSIPRF

Table 11. Defining and verifying operator authority (continued)

Keyword	**Related Defaults**	**Restrictions**	**Effect**
OPERSEC= SAFCHECK	• CMDAUTH must be specified.		• Password verification using SAF product • Operator must be defined in DSIOPF • Profile read from DSIPRF • NVSS task user IDs are used for any SAF calls for NVSS operators, such as to the DATASET class.
OPERSEC= SAFDEF	• CMDAUTH must be specified.		• Password verification using SAF product • Operator logon authority using RACF APPL class • Operator attributes defined in NetView segment of SAF product • NVSS task userids are used for any SAF calls for NVSS operators, such as to the DATASET class.

Protecting Commands:

Table 12 shows how the CMDAUTH and related options are used to specify how operator command authority checking is to be performed.

Table 12. Protecting commands executed in NVSS

Keyword	**Related Defaults**	**Restrictions**	**Effect**
CMDAUTH= TABLE	AUTHCHK default is SOURCEID.		Command authorization is based on the specified table.
CMDAUTH= SAF	AUTHCHK default is SOURCEID.	Cannot be specified if OPERSEC is MINIMAL or NETVPW.	Command authorization using the NETCMDS class of an SAF product. Immediate commands are not checked in the NETCMDS class, but a backup command authorization table can be used for this purpose.
TBLNAME= *table_name*		Required if CMDAUTH=TABLE	Identifies the table to be used

Table 12. Protecting commands executed in NVSS (continued)

Keyword	Related Defaults	Restrictions	Effect
BACKTBL= *table_name*		Valid only when CMDAUTH=SAF	Specifies the backup table to be used for immediate commands and when the SAF product cannot make a security decision. This can occur when: • There is no resource name defined in the NETCMDS class which protects or authorizes this command • The NETCMDS class is not active • The security product is not active.
SAFNODEC= PASS|FAIL		Defaults to PASS Only used when CMDAUTH=SAF and no BACKTBL is specified	Identifies whether to PASS or FAIL command authority checking if the SAF product can reach no decision.
AUTHCHK= SOURCEID	Default when CMDAUTH=TABLE	Authorization is based on the authority of the original issuer of the command or the ID closest to the original issuer.	
AUTHCHK= TARGETID			Authorization is based on the authority of the ID under which the command runs.

Usage Notes:

- When CMDAUTH=SAF and no backup command authorization table is used, immediate commands are not checked and will pass even if SAFNODEC=FAIL.

PROFILE

The PROFILE statement defines the profile name to the system. PROFILE must be the first statement in each profile definition. Code this statement in a member specified by a PROFILEN statement associated with the operator. See "OPERATOR" on page 205 and "PROFILEN" on page 214 to determine how a PROFILEN statement is associated with an operator. Profiles are not used when OPERSEC=SAFDEF. Examples of sample members that are supplied with the NVSS program are DSIPROFA and DSIPROFB, which can be found in NETVIEW.DSIPRF.

The syntax for the PROFILE statement is:

PROFILE

```
►►──profilename PROFILE──┬─────────────┬──┬────────────────────┬──┬──────────┬──►◄
                         └─HCL=hclname─┘  └─,CONSNAME=consname─┘  └─,IC=text─┘
```

Where:

profilename
Indicates the name of the member that contains the profile. This name must begin in column 1.

HCL=*hclname*
Indicates the name of the hardcopy printer that is automatically started when this operator logs on. Define this name in the VTAM definition and in the msys for Operations HARDCOPY definition statement in DSIDMN. HCL is an optional operand. The IC keyword, when specified, must always be specified as the last keyword.

Although each operator can be assigned to only one hardcopy printer, several operators can share the same printer. However, if too many operators share the same hardcopy printer, messages for that device can accumulate and messages might not be printed for some time after they are received.

CONSNAME=*consname*
Indicates the default extended console name for operators using this profile. This default console name is used when the operator does not specify a console name using the GETCONID or SETCONID command. It is also the console name used when you issue the MVS command and have not previously obtained an extended console. The console name must be a 2 to 8 characters long, as required by MVS. Valid characters for console names are A–Z, 0–9, @, #, or $. The first character of the console name must be alphabetic (A–Z) or one of the following special characters: @, #, or $. For more information on console names, refer to the MVS library. For more information on the implications of specifying CONSNAME, refer to the GETCONID and SETCONID command in the msys for Operations online help. The IC keyword, when specified, must always be specified as the last keyword.

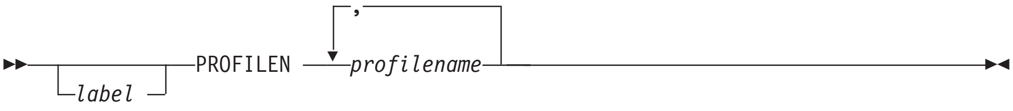
IC=*LOGPROF1***|***LOGPROF2LOGPROF3LOGPROF4*
Specifies the command list that is run immediately after a successful log on. Use LOGPROF1 for Msys for Operations operators, because it defines PF keys and a unique console name. Use LOGPROF2 for the AUTO1 autotask. LOGPROF3 assigns a unique console name to the task and should be used for the Msys for Operations autotasks other than AUTO1 and AUTO2. Use LOGPROF4 for the AUTO2 autotask.

PROFILEN

The PROFILEN statement associates the name of a particular profile or list of profiles with an operator identification. Code PROFILEN as often as necessary to ensure that all the possible profile names are associated with a particular operator identification. An OPERATOR statement must precede each PROFILEN statement or group of statements. You code this statement in DSIOPF. Profiles are not used when OPERSEC=SAFDEF.

The syntax for the PROFILEN statement is:

PROFILEN



Where:

label
: Indicates the optional label for the PROFILEN statement. This label identifies the statement in any related error messages.

profilename [,...]
: Indicates the profile name to be associated with the operator identification defined in the preceding OPERATOR statement. The profile name is a 1–8 character name that matches the *profilename* given on a PROFILE statement in a profile member. The first name listed in the first PROFILEN statement is used by default if an operator does not specify a *profilename* in the log on request. Note that multiple profile names must be separated by commas.

TRANSTBL

The TRANSTBL statement defines a character translation set to the NVSS program. Code this statement in the DSIDMN member. Stop and restart the NVSS program to implement the changes.

The syntax for the TRANSTBL statement is:

TRANSTBL

```
                           ┌─DSIEBCDC─┐
►►─┬─────────┬─TRANSTBL MOD=─┼──────────┼──────────────────────────────►◄
   └─label ──┘               ├─DSIEBCDC─┤
                             ├─DSIKANJI─┤
                             └─DSIKTKNA─┘
```

Where:

label
: Is the optional label for the TRANSTBL statement. This label identifies the statement in any related error messages.

MOD=DSIEBCDC|DSIKANJI|DSIKTKNA
: Specifies a particular load module name that contains a 1024-byte character translation set. Specify MOD=DSIEBCDC for EBCDIC support, MOD=DSIKANJI for kanji support, or MOD=DSIKTKNA for katakana support.

DSIEBCDC
: Selects an 8-bit coded character set called EBCDIC. This is the default.

DSIKANJI
: Selects a character set of symbols used in Japanese ideographic alphabets called kanji.

DSIKTKNA
: Selects a Japanese character translation set called katakana.

Usage Notes:

- All devices must use the same character set for meaningful results.
- The TRANSTBL value for the log printer program should be the same value as the value used for the NVSS program definition. The NVSS program does not check these values for compatibility.
- If you define more than one TRANSTBL statement, the NVSS program uses the last one you entered.

Part 6. Automation-related messages

This part provides a reference of new messages that have been introduced with APAR OW56107 and that are not yet accessible with LookAt (see "Using LookAt to look up message explanations" on page xii), and has the following chapter:

Chapter 17. Messages

AOF924A **AUTOMATION OF MESSAGE** *ixc_msg* **TERMINATED. REASON:** *reason***[, {PROCESSOR OPERATIONS | BCPII} COMMAND RC=***rc***]. SYSID:** *sysname***.**

Explanation: The variable *ixc_msg* shows the message that triggered the automation. It can have the following values and explanations:

IXC102A
XCF is removing a system from the sysplex.

IXC402D
XCF determined that a system in the sysplex appears to be inoperative.

XCF waits for the reply to proceed. However, the reply could not be automated.

The variable *reason* shows the reason code that was issued. It can have the following values:

10 The message being automated has not been formatted as expected. The message identifier could not be located.

11 A command was issued to the MVS system which reported that a system left the sysplex. A timeout occurred while waiting for a reply to this command.

12 The proxy resource name for the system leaving the sysplex could not be determined.

13 The Support Element of the system leaving the sysplex cannot be reached.

14 The target system name of the ISQ900I message could not be obtained or the target system which sent the ISQ900I message is not initialized to processor operations.

15 The replyid for the message being automated could not be determined.

16 The system name could not be located in the automated message.

17 The message that triggered the automation could not be retrieved for automation.

18 The automation requirements for the system leaving the sysplex could not be determined.

19 An error occurred while checking if the message was still outstanding.

30 A processor operations command failed. Refer to the appropriate command description.

31 A timeout occurred while waiting for the response of the Processor Management command.

32 The reply to the outstanding WTOR could not be sent.

33 An error occurred while determining the status of the local sysplex.

The variable *rc* shows the value of the return code.

System Action: Processing terminates.

Operator Response: Depending on which message triggered the automation, respond as follows:

IXC102A
Complete the shutdown of the system leaving the sysplex. Then reply DOWN to the outstanding WTOR.

IXC402D
Either reply 'INTERVAL=*sssss*' (range 0 to 86400) to give the system the specified interval to become operative again. Or, complete the shutdown of the system leaving the sysplex. Then reply DOWN to the outstanding WTOR.

System Programmer Response: Correct the problem. If reason code 11 was issued, no action is required.

Module: AOFRX700, AOFRX701, AOFRX702, INGRX705, INGRX706

Classes: 40, 43

TEC: YES

AOF925I **AUTOMATION OF MESSAGE** *ixc_msg* **FOR** *system* **COMPLETED SUCCESSFULLY**

Explanation: The automation of message IXC102A or IXC402D for system *system* ompleted successfully. The system is no longer part of the sysplex.

System Action: None.

Operator Response: None.

System Programmer Response: None.

Module: AOFRX700, AOFRX701, AOFRX702, INGRX705, INGRX706.

Classes: None.

TEC: NO

AOF926I **ERROR *error* DETECTED DURING AUTOMATION OF MESSAGE *ixc_msg***

Explanation: The routine responsible for the automation of messages IXC102A and IXC402D found an error.

- The variable *error* shows the error condition. The following error conditions can occur:

  10 The message is neither the IXC102A nor the IXC402D message.

  11 The same message is being processed by another task.

  12 Two or more commands were defined in the customization dialogs for message IXC102A. However, only one ISQCCMD can be issued.

  13 The command defined for IXC102A message is not an ISQCCMD command.

  14 The reply is no longer outstanding.

  15 Incorrect call of a subsequent clist.

  30 Using the supplied or default command, an attempt was made to deactivate the system leaving the sysplex. Another attempt will be made using the default command.

System Action: Processing terminates for conditions 10, 11, 14, and 15. For conditions 12, 13 and 30 processing continues using the default processor management command SYSRESET CLEAR.

Operator Response: None.

System Programmer Response: If error conditions 12, 13 or 30 occurred, correct the definitions and reload the automation control file.

Module: AOFRX700, AOFRX701, AOFRX702, INGRX705, INGRX706.

Classes: None.

TEC: NO

AOF960E **HARDWARE INFORMATION OF *system* COULD NOT BE VALIDATED.**

Explanation: The automation detected that neither the hardware information of the indicated system has been defined, nor a connection to the Support Element of the indicated system has been made for any of the registered systems. If the system fails, the automation (if enabled) is not able to take the appropriate hardware actions to prevent possible hardware-related outages caused by the system.

The following variables are used:

system The name of a system or a coupling facility.

System Action: Processing continues.

Operator Response: None.

System Programmer Response: Add the hardware definitions and make them available to all registered systems using the ACF command. You can ignore this message if: the indicated system will run the automation, you have defined the necessary hardware information, and the system is able to contact the hardware.

Module: INGRX804, INGRX809

Classes: 40.

TEC: NO

AOF961I **UNABLE TO CANCEL UN-NAMED JOB (SYSTEM=*sysname*/ASID=*asid*/TCB=*tcbaddr*). RESOURCE=*resource*.**

Explanation: Automation detected a long running enqueue but is unable to cancel the job because the job name is unknown and it is running on an un-automated system.

The following variables are used:

sysname
The name of the system running the address space.

asid The address space id running the task.

tcbaddr The TCB address of the task holding the enqueue.

resource The enqueue resource major and minor name.

System Action: None.

Operator Response: None.

System Programmer Response: Ensure the enqueue is released and terminate the job if necessary.

Module: INGRX741

Classes: 40, 43.

TEC: YES

AOF962I **UNABLE TO TERMINATE UN-NAMED JOB (SYSTEM=*sysname*/ASID=*asid*/TCB=*tcbaddr*). RESOURCE=*resource*.**

Explanation: Automation detected a long running enqueue but is unable to cancel the job because the job name is unknown. An attempt to abend the task has also failed. Message AOF200I will detail why the abend has failed.

The following variables are used:

sysname
The name of the system running the address space.

asid The address space id running the task.

tcbaddr The TCB address of the task holding the enqueue.

resource The enqueue resource major and minor name.

System Action: None.

Operator Response: None.

System Programmer Response: Ensure the enqueue is released and terminate the job if necessary.

Module: INGRX741

Classes: 40, 43.

TEC: YES

AOF963I UN-NAMED JOB (SYSTEM=*sysname*/ASID=*asid*/TCB=*tcbaddr*) IS BEING TERMINATED. RESOURCE=*resource*.

Explanation: Automation detected a long running enqueue but is unable to cancel the job because the job name is unknown. Automation is attempting to abend the task.

The following variables are used:

sysname
The name of the system running the address space.

asid The address space id running the task.

tcbaddr The TCB address of the task holding the enqueue.

resource The enqueue resource major and minor name.

System Action: None.

Operator Response: None.

System Programmer Response: Check that the job has abended and ensure that the enqueue is released.

Module: INGRX741

Classes: 40, 43.

TEC: YES

AOF964I Due to the detection of a long *minor_res* lock the task *taskid* in address space *asid* on system *sysname* is being abended.

Explanation: The automation detected a lock on the indicated minor system resource being held for more than 10 seconds. To prevent the lockout of further commands the task holding the lock is being abended with the ability to do its own recovery.

The following variables are used:

minor_res The minor resource name of the lock. The major name is SYSIEFSD.

taskid The task id holding the lock.

asid The address space id running the task.

sysname The name of the system running the address space.

System Action: The automation calls the Recovery Termination Manager to abend the indicated task.

Operator Response: None.

System Programmer Response: None.

Module: INGRX743

Classes: 40, 43.

TEC: YES

AOF965I The command *command* issued by *jobname* is being purged due to a hung command detection.

Explanation: The automation detected a command that is still executing. The command is abended to avoid lockouts of other commands.

The following variables are used:

command The first two words of the command text when applicable.

jobname The job name issued the command.

System Action: The automation abends the indicated command.

Operator Response: None.

System Programmer Response: None.

Module: INGRX743

Classes: 40

TEC: NO

AOF966I Value *value* of type *type* for *sysname* could not be evaluated.

Explanation: The value for the indicated type could not be verified. The reason is that the BCP internal interface to the corresponding Support Elements could not be established on any of the registered systems in the sysplex.

The following variables are used:

sysname The name of the defined operating system.

value The defined value.

type The type in question. This can be one of the following:

CPC The CPC name.

LPAR The LPAR name.

SYSPLEX The SYSLEX name.

TYPE The operating system type, such as MVS or CF.

System Action: Processing continues.

Operator Response: None.

System Programmer Response: None.

Module: INGRX809

Classes: 40

TEC: NO

AOF967E Value mismatch detected between *system1* **and** *system2* **for** *system3* **and type** *type***.**

Explanation: The automation detected that the value of the indicated type could not be verified. The reason is that the BCP internal interface to the corresponding Support Elements wasn't established on any of the registered systems in the sysplex. In addition, at least two different definitions exist for the indicated type on different msys systems.

As soon as the automation gets access to the Support Element, the value will be re-evaluated and automatically corrected.

The following variables are used:

system1 The name of a system running msys for Operations.

system2 The name of a system running msys for Operations.

system3 The name of the defined operating system.

type The type in question. This can be one of the following:

CPC The CPC name.

LPAR The LPAR name.

SYSPLEX The SYSLEX name.

TYPE The operating system type, such as MVS or CF.

System Action: Processing continues.

Operator Response: None.

System Programmer Response: Check and correct the definitions before the next start-up of the affected systems.

Module: INGRX809

Classes: 40

TEC: NO

AOF968E Value *oldvalue* **of type** *type* **for** *sysname1* **on** *sysname2* **has been replaced by** *newvalue***.**

Explanation: The automation detected that the user's hardware definition differs from the actual hardware. If the local system name is not shown in the message, a different setup other than the local system has been used . This should generally be avoided.

The following variables are used:

sysname1 The name of the defined operating system.

newvalue The new value.

oldvalue The improper value. This may be ***(NULL)*** if a value has been found in the hardware configuration but there is no definition in AOFCUST.

sysname2 The name of the system where the improper setting was detected.

type The type in question. This can be one of the following:

CPC The CPC name.

LPAR The LPAR name.

SYSPLEX The SYSLEX name.

TYPE The operating system type, such as MVS or CF.

System Action: The improper definition is temporarily replaced by the actual value to prevent any outage that could be caused by the old value.

Operator Response: None.

System Programmer Response: Check and correct the definitions in AOFCUST before the next start-up of the indicated system.

Module: INGRX809

Classes: 40

TEC: NO

HSAM5211I MEMBER *member* **FOUND IN** *dsn***.**

Explanation: The indicated member has been found in the indicated data set of the PARMLIB concatenation.

The variable *member* shows the name of the member being processed using the MVS PARMLIB service.

The variable *dsn* shows the data set name from which the member has been read.

System Action: None.

Operator Response: None.

System Programmer Response: None.

Module: HSAPSIPL

HSAM5212E I/O ERROR READING MEMBER *member*.

Explanation: An I/O error occurred when trying to read the indicated PARMLIB member.

The variable *member* shows the name of the member being processed using the MVS PARMLIB service.

System Action: None.

Operator Response: Inform your system programmer.

System Programmer Response: Correct the problem and re-run the program.

Module: HSAPSIPL

ING805I *requestor* REQUESTS TO CONNECT TO CPC *cpcaddr*.

Explanation: The hardware interface tries to establish a connection to the indicated processor hardware.

The variable *requestor* shows the originator of the request.

The variable *cpcaddr* shows the address of the CPC.

System Action: Processing continues.

Operator Response: None.

System Programmer Response: None.

Classes: 40.

Module: INGRVX80

ING806E COMMUNICATION WITH CPC *cpcaddr* CANNOT BE ESTABLISHED

Explanation: A failure occurred while the processor hardware was contacted through the hardware interface.

The variable *cpcaddr* shows the address of the CPC with which no session could not be established.

System Action: None.

Operator Response: None.

System Programmer Response: Check the netlog for AOFA*xxxx* messages to obtain more information, for example the available condition and sense codes. Correct the problem and restart the session to the PC.

ING810I *requestor* REQUESTS TO DISCONNECT FROM CPC *cpcaddr*.

Explanation: The hardware interface terminates the communication with the indicated processor hardware.

The variable *requestor* shows the originator of the request.

The variable *cpcaddr* shows the address of the CPC.

System Action: Processing continues.

Operator Response: None.

System Programmer Response: None.

Classes: 40.

Module: INGRVX80

ING910I HEALTH CHECKER BACKEND TASK IS ACTIVE

Explanation: The customer has turned on the health checking function. This causes the NetView task that runs the health checker backend (AOFHC) to be started.

System Action: The health checking function is activated.

Operator Response: None.

System Programmer Response: None.

Classes: None.

ING911I HEALTH CHECKER BACKEND TASK HAS TERMINATED

Explanation: The customer has turned off the health checking function. This causes the NetView task that runs the health checker backend (AOFHC) to be terminated.

System Action: The health checking function is deactivated.

Operator Response: None.

System Programmer Response: None.

Classes: None.

ING912E ENVIRONMENT PROBLEM WITH HEALTH CHECKER BACKEND, REASON:

Explanation: The NetView task running the health checker backend is either not active or has detected some error situation in the run time environment.

System Action: The requested function cannot be performed. If the task is not already inactive, the task might terminate depending on the severity of the problem.

Operator Response: Contact the system programmer.

System Programmer Response: Try to analyze the problem to see whether there might be a shortage of system resources. If you cannot resolve the problem, please call IBM support.

Classes: None.

ING913I ERROR IN LINE *x* POSITION *y* IN USERPARM FILE

Explanation: This prefix is used for messages of the parser of the health checker function. The cause of the problem is some syntax error in the customer's override of IBM's best practices.

This message extends over multiple lines, the follow on lines have message prefix ING917I.

System Action: The health checker continues, the override in error is not ignored. This means that the check with the erroneous override is not performed at all until the problem is fixed.

Operator Response: Contact the system programmer.

System Programmer Response: The reason explains the cause of the error as well as the check which detected the error. Please use this information to correct the problem.

Please note that the line number does not refer to your original data. Instead, this message refers to the NetView global variables AOF.0INGPKMAI.*.

Please use command 'qryglobl vars=AOF.0INGPKMAI.*' to display the values of these variables to analyze the problem.

Go back to the original definition of your overrides, locate the line with the error, and correct it.

The line you identified with this procedure is the line which triggered the parser to detect the error. The actual line of error could be before this line.

Classes: None.

ING914E CUSTOMIZATION ERROR FOR HEALTH CHECKER BACKEND, REASON:

Explanation: The NetView task running the health checker backend has detected some error situation in your user overrides of IBM's best practices.

If this message extends over multiple lines, the follow on lines have message prefix ING917I.

System Action: The health checker continues, the override in error is not ignored. This means that the check with the erroneous override is not performed at all until the problem is fixed.

Operator Response: Contact the system programmer.

System Programmer Response: The reason explains the cause of the error as well as the check which detected the error. Please use this information to correct the problem.

These messages may refer to 'Line x in file USERPARM'. In such case the reference actually means the NetView variable AOF.0INGPKMAI.x.

Please use command 'qryglobl vars=AOF.0INGPKMAI.*' to display the values of these variables to analyze the problem.

Go back to the original definition of your overrides, locate the line with the error, and correct it.

Classes: None.

ING915I EXCEPTION DETECTED:

Explanation: This message is used for automation reasons. It is issued when the Health Checker function detects an exception during a check. The name and severity of the particular check that raised the exception is shown in the message text.

System Action: None.

Operator Response: None.

System Programmer Response: Use command INGHC to look at the report of the Health Checker function to determine the cause of the exception.

Classes: 40, 46.

ING916I *text*

Explanation: These messages are used internally by the health checker function and would normally not be visible on the console or in the netlog. The messages have varying contents, they are used to send data back and forth between frontend and backend.

System Action: None.

Operator Response: None.

System Programmer Response: None.

Classes: None.

ING917I *text*

Explanation: This message is used for formatting reasons. It extends other messages with prefix 'ING91*x*' if these messages get longer than one line.

Please use the help for the message with prefix 'ING91*x*' preceeding this message.

System Action: None.

Operator Response: None.

System Programmer Response: None.

Classes: None.

ING918I BEST PRACTICES POLICIES ARE UNAVAILABLE

Explanation: You requested the best practices policy information for the health checker function.

However, the system from where the information is to be retrieved currently does not run the health checker function.

If you did not explicitly specify a system name, this message means that currently no system in the sysplex runs the health checker function.

System Action: Processing terminates.

Operator Response: Run the command 'INGAUTO ON HEALTHCHK' on at least one system in the sysplex and repeat your request.

System Programmer Response: None.

Classes: None.

ING919E HEALTH CHECKER FUNCTION DISABLED

Explanation: In order to correctly apply the specified filters for the display of the Health Checker results, command INGHC must communicate with the Health Checker backend task. This message occurs when this communication is not successful.

Most probably this message means that the Health Checker function is disabled, you may use command 'DISPFLGS' to verify this.

System Action: Processing continues.

Operator Response: Run the command 'INGAUTO ON HEALTHCHK' on the system where you want to run the INGHC command and press PF9 to refresh the display.

System Programmer Response: None.

Classes: None.

Part 7. Appendixes

This part contains the following appendices:

- Appendix A, “Making security definitions using the command authorization table,” on page 229
- Appendix B, “Return codes,” on page 255
- Appendix C, “Coexistence of msys for Operations and SA OS/390 releases,” on page 265
- Appendix D, “msys for Operations customization checklist,” on page 267
- Appendix E, “The IBM Health Checker for z/OS and Sysplex checks,” on page 331
- Appendix F, “Response messages, error strings, condition codes,” on page 341
- Appendix G, “Sense codes, hardware object status summary,” on page 361

Appendix A. Making security definitions using the command authorization table

Defining security using msys for Operations definitions

Use the information in this appendix to define your security using only msys for Operations definitions instead of using an SAF product. msys for Operations delivers a INGSCAT1 sample file. The sample predefines security settings for your use. To use the sample, copy it to your DSIPARM concatenation, customize it there and change the OPTIONS statement in DSIDNMK to `CMDAUTH=TABLE,TABLENAME=INGSCAT1`. The following assumes that you either tailor the file to your enterprise's requirements, or that you issue the necessary commands from the system console.

Note that the HSAET32 BCP interface requires the use of an SAF product. If you plan to use the CF ENABLE, CF DRAIN, and IXC102A automation functions, it is recommended to use an SAF-related OPERSEC value. MINIMAL or NETVPW is not recommended.

Overview of operator security

Operator security can be defined using msys for Operations, an SAF product such as RACF, or a combination of both. The OPERSEC keyword in the OPTIONS statement determines which type of operator security is used. Use the REFRESH command to change the value of the OPERSEC keyword.

Review the information in Table 13 to determine which type of operator security best meets your requirements.

Note: It is recommended to use an SAF product exclusively (OPERSEC=SAFDEF) for operator security.

Table 13. Operator security definition types

OPERSEC value	Type of operator password and logon attributes
MINIMAL	Both operator passwords and logon attributes are ignored.
NETVPW	Operator passwords and logon attributes are provided exclusively by msys for Operations. Operator passwords are specified in DSIOPF. Logon attributes are specified in member DSIOPF and defined in DSIPRF.
SAFDEF	Operator passwords and logon attributes are provided exclusively by an SAF product. Operator passwords are checked by an SAF product, and logon attributes are defined in the NetView segment of an SAF product. Access to the data sets protected in the DATASET class and to MVS system commands protected in the OPERCMDS class of the SAF product are checked at the individual task level.
SAFCHECK	Operator passwords and logon attributes are provided by a combination of msys for Operations and an SAF product. Operator passwords are checked by an SAF product, with operator profiles specified in msys for Operations member DSIOPF and logon attribute values defined in DSIPRF. Access to the data sets protected in the DATASET class and to MVS system commands protected in the OPERCMDS class of the SAF product are checked at the individual task level.

For a description of the OPERSEC keyword, see Chapter 16, “msys for Operations definition statements reference,” on page 203.

By defining msys for Operations operators exclusively to an SAF product, you eliminate the need for members DSIOPF and DSIPRF. See “Defining operators, passwords, and logon attributes” on page 75 for more information on defining operators and operator attributes using an SAF product.

Operator identifiers

Define a unique operator identifier for each operator who logs on to msys for Operations.

Do not use the names of msys for Operations commands, components, printers (hardcopy logs), terminals, or task identifiers for operator identifiers. Also, do not use the following reserved keywords:

ALL
DPR
DST
HCL
HCT
LOG
MNT
NNT
OPT
OST
PPT
SYSOP
TCT

Using msys for Operations for password authorization

Use password security to prevent unauthorized logging on to msys for Operations. To use msys for Operations for password authorization, specify OPERSEC=NETVPW on the OPTIONS statement in DSIDMN or on the msys for Operations REFRESH command. The password stored in DSIOPF is used to check logon password authorization, so you must update DSIOPF to change a password.

Note: To prevent unauthorized viewing or modification of DSIOPF and command lists which contain passwords, see “Controlling access to data sets and members” on page 251.

Define the operator identifier and password with the OPERATOR definition statement in DSIOPF as follows:

```
NEWOPER   OPERATOR   PASSWORD=NEWOPER
          PROFILEN   DSIPROFB
```

Where NEWOPER is the operator identifier and NEWOPER is the operator password.

Using msys for Operations without password authorization

When OPERSEC=MINIMAL is defined in the OPTIONS statement, msys for Operations does not perform any password checking. Unless you use other ways of keeping your system secure, such as physically restricting access to terminals, you should use password security.

Operator logon attributes

Operator logon attributes describe characteristics associated with an operator.

Operator logon attributes can be defined in msys for Operations, in an SAF product, or in both. Although only one definition can be in effect at a time, you can dynamically change whether operator logon attributes are used from msys for Operations operator profiles (DSIPRF) or the NetView segment of an SAF product.

Whether you define operator profiles in DSIPRF or define operators in an SAF product, altering the logon attributes will not have an effect on the task until it is logged off, then logged on again. Before altering or migrating operator definitions, you should understand the following operator attributes:

- IC keyword on a PROFILE statement
- MSGRECVR keyword on an AUTH statement

Using MSGRECVR

The MSGRECVR attribute of the AUTH statement can be used in both msys for Operations operator profiles and in the NetView segment of an SAF product. It specifies whether operators are eligible to receive unsolicited messages that are not routed to a particular operator using either the msys for Operations ASSIGN command or msys for Operations automation.

For more information about using the MSGRECVR keyword in a DSIPRF profile, see to “AUTH” on page 204.

Using IC

If the IC keyword is specified, it must be the last keyword on the PROFILE statement. For more information about using the IC keyword in a DSIPRF profile, see “PROFILE” on page 213.

Defining operator attributes in msys for Operations profiles

You can code more than one profile for an operator. You can also use the same profile for more than one operator. For each operator profile, create a profile member in DSIPRF with a PROFILE definition as the first statement in that file. Other definition statements, such as AUTH, follow this PROFILE statement.

Here is an example showing how you could add an operator definition to the DSIOPF member of DSIPARM:

```
NEWOPER   OPERATOR  PASSWORD=NEWOPER
          PROFILEN  DSIPROFA
```

You can define profiles that:

- Specify a command or a command list to run automatically when an operator logs on
- Specify whether an operator is eligible to be the authorized receiver of undeliverable messages.

For examples of profile definitions, browse profiles DSIPROFA and DSIPROFB.

Here is an excerpt of sample profile DSIPROFA:

```
DSIPROFA  PROFILE  IC=LOGPROF1
          AUTH    MSGRECVR=NO,CTL=GLOBAL
                  END
```

Note: msys for Operations does not use the CTL statement defined in DSIPROFA.

The profile in the previous example specifies:

IC=LOGPROF1
A command list named LOGPROF1 (CNME1049) is run automatically when an operator logs on with this profile.

MSGRECVR=NO
Operator is not eligible to be the authorized receiver.

You can define other profiles as necessary by creating additional profile members in DSIPRF. For more information about creating profile members, see “PROFILE” on page 213.

Dynamically adding or deleting operators

If you are using msys for Operations operator definitions in DSIOPF, you can use the msys for Operations REFRESH OPERS command to dynamically add or delete operators while msys for Operations is running. This command refreshes operator definitions in DSIOPF that were added since the last time msys for Operations was stopped and restarted or since the last REFRESH OPERS command was issued. You can also use the `REFRESH OPERS,TEST` command to check the operator definitions that will change when you issue the REFRESH OPERS command.

Command authorization

This section provides the following information about command authorization:

- An overview of command authorization
- Command authorization using the command authorization table
- Command authorization for specific commands

It is recommended that you review this section in its entirety before you begin to define your command security.

Overview

Command authorization is the process of protecting commands from unauthorized use.

A ***command authorization table*** (CAT) enables you to restrict access to commands, keywords, and values. It then allows you to permit operators and groups of operators to access these restricted commands, keywords, and values. You can also specify commands, keywords, and values that pass authorization checking. The NVSS command authorization table is stored as a member of DSIPARM. You can use the REFRESH command to dynamically update your table. See “Command authorization using a command authorization table (CAT)” on page 234 for more information.

Recommended commands to protect: Customers must decide which commands to protect based on their unique security requirements. However, it is recommended that, at a minimum, you restrict the following commands, because they can affect the msys for Operations environment or access to it:

- AFTER (Use of the PPT keyword.)
- AT (Use of the PPT keyword.)
- AUTOTBL
- CHRON (Use of the ROUTE keyword. See “Defining security for the CHRON command” on page 247 for more information.)
- CLOSE
- DEFAULTS
- EVERY (Use of the PPT keyword)
- EXCMD. (See the msys for Operations online help for more information.)

- FOCALPT
- GETCONID
- GLOBALV
- INGRCCHK
- MODIFY
- MVS (See “MVS” on page 128 for more information.)
- OVERRIDE
- PURGE
- READSEC (See the online help for more information.)
- REFRESH
- RMTCMD
- RUNCMD
- SUBMIT (See the online help for more information.)
- SETCONID
- START
- STOP
- VARY
- WRITESEC (See the online help for more information.)

Exceptions to command authorization checking: Major exceptions to command authorization checking include:

- Commands entered as replies to the msys for Operations WTOR (message DSI802A) are not authority checked. To prevent users from issuing commands using the WTOR, specify CMDWTOR=NO in the MVSPARM statement in DSIDMN. This prevents msys for Operations from issuing the WTOR.
- Command authority checks are not made against the PPT or DST tasks; therefore, you do not need to authorize these tasks to access your protected commands.
- Commands issued from a source ID of `*BYPASS*` are not checked for command authorization by The NVSS command authorization table. The SOURCEID will default to `*BYPASS*` if the command was entered at an extended multiple console support (EMCS) console and the operator was not logged on to the EMCS console.

Auditing command authority checking: You can audit accesses to protected commands, keywords, and values. This auditing can be done on an individual command, keyword, or value basis.

For command authorization using a NVSS command authorization table, you can specify the CATAUDIT keyword on the DEFAULTS command to determine the level of auditing performed. With the option not to audit, you can chose to audit unsuccessful or failed attempts to access protected command identifiers, or to audit all matches on command identifiers in your table. For more information on the DEFAULTS command, refer to the NVSS online help. You can also specify specific auditing levels on specific command identifiers using PROTECT and EXEMPT statements with the AUDIT keyword, as described in “Table Statements” on page 236.

If auditing is specified, the records are written to SMF as record type 38.

Protecting commands containing special characters: There are some special characters that cannot be included in the command identifier or SAF resource name. For this reason, NVSS translates these special characters to other characters before passing them to either the NVSS command authorization table. The special characters that are translated along with their translated results are:

Reserved Character	Translated Result
.	/
*	+
%	?
&	:
- (dash)	_ (underscore)
' ' (blank)	_ (underscore)

As an example, the following NVSS command can be entered by a NVSS operator:

```
LIST MEMSTAT=*
```

To restrict access to the FPCAT keyword and its value in the command authorization table, include the following statement:

```
PROTECT NETA.MSO01.LIST.MEMSTAT.+
```

Note that the asterisk was translated to a plus.

Command authorization using a command authorization table (CAT)

NVSS provides the ability to use a NVSS command authorization table to restrict the use of commands and operands to specific operators or groups of operators. The table consists of a member in DSIPARM containing the authorization statements. This table can include statements to embed other members from DSIPARM. Using a NVSS command authorization table, you can also protect command lists that do not have a CMDMDL statement in DSICMSYS. The sample command authorization table supplied with msys for Operations is INGSCAT1. The following sections provide the information that you need to modify INGSCAT1 to meet your security requirements.

Command identifiers: Identify which commands, keywords, and values are protected using command identifiers. The format of the command identifiers for the NVSS command authorization table is the same as the format of the resource names used in the NETCMDS class in an SAF product. In its full form, a command identifier uses these fields: *netid.luname.command.keyword.value*.

Use the NVSS LISTVAR command to determine the *netid* and *luname* values for your systems. In the LISTVAR example shown in Figure 57, the current values are NETA for *netid* and the NVSS domain, MSO01, for *luname*.

```
LISTVAR
CNM353I LISTVAR : OPSYSTEM = MVS/ESA
CNM353I LISTVAR : MVSLEVEL = SP5.1.0
CNM353I LISTVAR : CURSYS   = VTAM430
CNM353I LISTVAR : VTAMLVL  = VT43
CNM353I LISTVAR : VTCOMPID = 5695-11701-301
CNM353I LISTVAR : NETVIEW  = NV31
CNM353I LISTVAR : NETID    = NETA
CNM353I LISTVAR : DOMAIN   = MSO01
CNM353I LISTVAR : APPLID   = MSO01007
CNM353I LISTVAR : OPID     = OPER3
CNM353I LISTVAR : LU       = A01A703
CNM353I LISTVAR : TASK     = OST
CNM353I LISTVAR : NCCFCNT  = 0
CNM353I LISTVAR : HCOPY    =
CNM353I LISTVAR : CURCONID =
CNM353I LISTVAR : DATE     = 11/03/94
CNM353I LISTVAR : TIME     = 13:41
```

Figure 57. Example of LISTVAR command output

Note that some characters are reserved if you are using the NVSS command authorization table or an SAF security product for command authorization checking.

The command identifier can be up to 246 characters in length, including the periods that serve as field delimiters. The individual fields of the command identifier have no maximum length as long as the entire command identifier length does not exceed 246 characters.

You can use generic characters in command identifiers. An asterisk (*) can be used to indicate that all possible values of a field are protected or permitted, except those that are more explicitly specified. You can use the asterisk either as a replacement for a field or as a trailing character to indicate that all items that begin with the specified characters are to be protected. The percent sign (%) can be used as a single character generic anywhere within the command identifier. Generic characters are useful to specify a level of protection for commands for which there is not a match in the table. You can do this because the most specific command identifier determines the level of protection for a command.

Note: The generic character combination `%*` is not valid.

Commands are checked separately from keywords and values. When designing command identifiers, keep in mind that the command is checked first, in addition to the subsequent security checking for the command, keyword, and value combinations. Keywords and their associated values are checked as a pair. To protect a keyword that has a value associated with it, there must be an entry in the value position of the command identifier. The command identifiers can be in these formats:

```
netid.luname.command
netid.luname.command.keyword (used only for keywords without values)
netid.luname.command.keyword.value
```

Where:

netid
: Indicates the VTAM network identifier. You can specify a generic character (*) for this field.

  The *netid* specification is syntax checked for format (*netid* may not begin with a left parenthesis) but no checking is done to verify that the *netid* specified matches the current *netid*. This field is treated as a place holder and is supported so that the format of the command identifier in the NVSS command authorization table is the same as the format of a resource name in the NETCMDS class of an SAF product.

luname
: Indicates the domain identifier. Only statements which match your *luname* are loaded when the NVSS command authorization table is activated, but all statements are syntax checked, regardless of luname.

command
: Indicates the command name on the CMDMDL statement in the DSICMSYS member of DSIPARM, or a command list name. This must be the actual command name and not a synonym defined by the CMDSYN statement. No checking is done to validate that *command* is a valid command or command list name.

keyword
: Indicates the keyword identifier that is protected.

value
Indicates the value identifier that is protected when used with the keyword on the command.

Command authorization table syntax: Table statements consist of free-form text which specify a table statement type followed by its operands. You can enter the text in upper or lowercase, with the exception of %INCLUDE statements, which must be in uppercase. For all other statements, the text is converted to uppercase when the table is processed. The table statements must be coded between columns 1 and 72. If a statement is too long to fit between columns 1 and 72, you can use the <BEGIN> and <END> statements when multiple lines should be treated as a single statement. You can include a sequence number in columns 73 through 80 for problem determination purposes. If NVSS encounters any errors while processing the table statements, the error messages issued include the sequence number of the line in error. An asterisk in column 1 denotes a comment and causes the rest of the NVSS command authorization table line to be ignored.

Table Statements: This section describes the format and function of the following statement types:
- <BEGIN> and <END>
- %INCLUDE
- PROTECT
- EXEMPT
- GROUP
- PERMIT
- SETVAR

<BEGIN> and <END> Statements: The <BEGIN> and <END> statements specify the beginning and end of a NVSS command authorization table statement that spans multiple input lines. The total length of any individual table statement must not exceed 4096 characters, including blanks, which provides a maximum of 56 input lines.

The syntax for the <BEGIN> and <END> statements are:

<BEGIN>

```
►►──<BEGIN>──────────────────────────────────────────────────────────►◄
```

<END>

```
►►──<END>────────────────────────────────────────────────────────────►◄
```

The <BEGIN> and <END> statements must appear on lines by themselves. Command identifiers may continue onto more than one line, but statement types, group names, and each user ID in a userid_list should not span more than one line. As you enter multiple input lines, be careful not to accidentally put an asterisk in column 1, because the remainder of that line will be treated as a comment.

Example:

To enter a command authorization statement that spans two input lines, use the following:

```
<BEGIN>
GROUP ALLOPS OPER1,OPER2,OPER3,OPER4,OPER5,OPER6,OPER7,OPER8,OPER9,
             OPER10,OPER11
<END>
```

Note: Blank characters between input lines alignment are valid.

%INCLUDE statement: The %INCLUDE statement enables you to keep portions of your NVSS command authorization table in separate DSIPARM members. Both the %INCLUDE statement and its values (either the *membername* or *&varname*) must be capitalized.

The syntax for the %INCLUDE statement is:

%INCLUDE statement

```
►►──%INCLUDE ──┬─membername─┬──────────────────────────────────────────────►◄
               └─&varname───┘
```

Where:

%INCLUDE
Indicates the keyword coded at the beginning of each %INCLUDE statement.

membername
Indicates the name of the DSIPARM member to be included.

&varname
Indicates the name of an existing local or global variable, preceded by the ampersand (&) character.

Usage notes:

1. Each %INCLUDE statement can be no longer than one line.
2. A member that has been included can contain %INCLUDE statements as well as other NVSS command authorization table statements.
3. A member that has been included cannot include itself either directly or indirectly.
4. If you specify a variable name for the value of the %INCLUDE, the NVSS program includes the designated member when you issue the REFRESH command with CMDAUTH=TABLE. You cannot use a variable name in a command authorization table specified on an OPTIONS statement in DSIDMN for NVSS initialization. NVSS searches for the variables in the following order:
   - If the REFRESH command is issued from a command procedure, the NVSS program searches first for a local variable of the name *varname*, then for a task global variable, and finally for a common global variable.
   - If the REFRESH command is not issued from a command procedure, the NVSS program searches for a task global variable of the name *varname* and then for a common global variable.

   If you change the value of the variable after activating the NVSS command authorization table, the member that is included does not change, unless you reissue the REFRESH command.

Example:

To include member TBL02 from DSIPARM, include the following statement in your NVSS command authorization table:

```
%INCLUDE TBL02
```

PROTECT statement: The PROTECT statement identifies a command identifier to be protected.

The syntax for the PROTECT statement is:

PROTECT

```
►►──PROTECT ─┬──────────────────────────┬─command_identifier──────────────────────►◄
             └─(AUDIT=─┬─ALL──────┬─)─┘
                       ├─FAILURES─┤
                       └─NONE─────┘
```

Where:

command_identifier
Specifies the *netid, luname,* command, keyword, and value to be protected. See "Command identifiers" on page 234 for information on specifying command identifiers.

AUDIT
Specifies whether an audit record should be created when a command authority check yields a match on the command identifier. The audit records can be SMF type 38 records, or the DSIXITXL exit can write the records to an external log. The AUDIT keyword is optional. If not specified, auditing is determined by the value of CATAUDIT on the DEFAULTS command. When specified, the value overrides the value specified for CATAUDIT on the DEFAULTS command. Valid values for AUDIT are:

ALL
Specifies that an audit record is to be created when a match occurs on the command identifier.

FAILURES
Specifies that an audit record is to be created when a match occurs on the command identifier and the command authority decision is *fail*.

NONE
Specifies that no audit record is to be created when a match occurs on the command identifier.

Example:

To define a command identifier to protect the AUTH keyword and MASTER value of the GETCONID command in domain MSO01, use the following statement:

```
PROTECT *.MSO01.GETCONID.AUTH.MASTER
```

To define a command identifier to protect the AUTH keyword and MASTER value of the GETCONID command in domain MSO01, and to create audit records for all attempts to get a console with master authority, use the following statement:

```
PROTECT (AUDIT=ALL) *.MSO01.GETCONID.AUTH.MASTER
```

Usage notes:

- Create one PROTECT statement for each command that you want to protect. For example, to protect the STOP command for *luname* MSO01, create a table entry as follows:
  ```
  PROTECT *.MSO01.STOP
  ```
- Create one PROTECT statement for each command and keyword that does not have an associated value which you want to protect. For example, to protect the OFF keyword on the AUTOTBL command for *luname* MSO01, create a table entry as follows:
  ```
  PROTECT *.MSO01.AUTOTBL.OFF
  ```
- Create one PROTECT statement for each command, keyword, and value combination that you want to protect. For example, to protect the FORCE keyword with a value of AUTO1 on the STOP command for *luname* MSO01, create a table entry as follows:
  ```
  PROTECT *.MSO01.STOP.FORCE.AUTO1
  ```
- To protect all values of FORCE in the previous example, create a table entry as follows:
  ```
  PROTECT *.MSO01.STOP.FORCE.*
  ```
- To protect all values of FORCE that begin with ″TEST″ and end with ″0″, create a table entry as follows:
  ```
  PROTECT *.MSO01.STOP.FORCE.TEST%0
  ```
- To allow an NVSS operator to issue an NVSS command that is protected with a PROTECT statement, you must use a PERMIT statement for each operator ID or group of operators that should be authorized.
- If you have more than one statement that describes the same command, keyword, and value, the first is used and all others are ignored. The *netid* and *luname* values are ignored once the NVSS command authorization table is loaded. The following example shows how generic characters cause the second command identifier to be ignored. If the following statements are included in the NVSS command authorization table for domain MSO01, only the first is used:
  ```
  PROTECT *.*.AUTOTBL.MEMBER.DSITBL01
  PROTECT *.MSO01.AUTOTBL.MEMBER.DSITBL01
  ```

EXEMPT Statement: The EXEMPT statement identifies a command and optionally a keyword and value to be exempted from command authorization.

It enables all users to issue a command, keyword, or value, which is similar to defining a resource in the NETCMDS class with a universal access of read (UACC(READ)).

Using specific EXEMPT statements can reduce the amount of processing required for command authorization checking, and can improve performance.

The syntax for the EXEMPT statement is:

EXEMPT

```
►►──EXEMPT ──┬─────────────────────┬─command_identifier──────────────►◄
             └─(AUDIT=─┬─ALL──┬─)─┘
                       └─NONE─┘
```

Where:

command_identifier
Is the identifier specifying the *netid*, *luname*, command, keyword, and value to be exempted. See “Command identifiers” on page 234 for information on specifying command identifiers.

AUDIT
Specifies whether an audit record should be created when a command authority check yields a match on the command identifier. The audit records can be SMF type 38 records, or the DSIXITXL exit can write the records to an external log. The AUDIT keyword is optional. If not specified, auditing is determined by the value of CATAUDIT on the DEFAULTS command. When specified, the value overrides the value specified for CATAUDIT on the DEFAULTS command. The values allowed for AUDIT are:

ALL
Specifies that an audit record is to be created when a match occurs on the command identifier.

NONE
Specifies that no audit record is to be created when a match occurs on the command identifier.

Example:

To define a command identifier to exempt the LIST command in any domain, you must use the following statements:

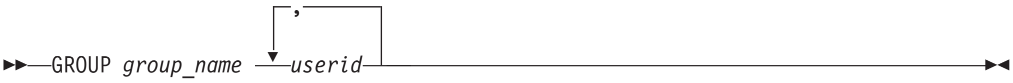
```
EXEMPT *.*.LIST
EXEMPT *.*.LIST.*
```

The first statement applies only to the LIST command. The trailing asterisk in the second statement causes this command identifier to apply to all keywords and values of the LIST command that are not more explicitly specified.

GROUP statement: The GROUP statement defines a list of operators to be associated with a specific group name for command security purposes. The group name is unrelated to other groups of operators, such as the groups used to route messages using the NVSS ASSIGN command.

The syntax for the GROUP statement is:

GROUP



Where:

group_name
Is the 1–8 character name of the group you are defining. The *group_name* cannot contain an ampersand (&), asterisk (*), or percent sign (%). The group name cannot be the same as any of your user IDs that are defined in the NVSS command authorization table.

userid
Is the 1–8 character identifier of a user to be included in the group. The *userid* cannot contain an ampersand (&), asterisk (*), or percent sign (%). This must be an individual user ID and not the name of a group.

Example:

To define a group named NIGHTOPS containing operators FELIX, MORRIS, and TOM, use the following:

```
GROUP NIGHTOPS FELIX,MORRIS,TOM
```

To define a large number of operators to a group, you can either repeat the same group name on multiple group statements or create a multiple-line group statement using the NVSS <BEGIN> and <END> statements.

PERMIT statement: The PERMIT statement authorizes a user ID or group to issue a command and optionally a keyword and value. The command identifier must have been previously protected with a PROTECT statement. You can include more than one PERMIT statement for the same command identifier.

The syntax for the PERMIT statement is:

PERMIT

```
►►──PERMIT authorized_name command_identifier──────────────────────────────►◄
```

Where:

authorized_name
Is the 1–8 character name of a user ID or a group that is authorized to issue the command, keyword, and value identified by the *command_identifier*. The *authorized_name* cannot contain an ampersand (&), asterisk (*), or percent sign (%). No checking is done to verify that a user ID is a valid NVSS operator ID.

Note: User IDs used in your table statements are independent of DSIOPF operator definitions and SAF product definitions. Even if an operator has been deleted from DSIOPF or the SAF product, the operator will continue to have the same command authority with respect to the active NVSS command authorization table as long as the operator remains logged on.

command_identifier
Is the identifier specifying the *netid, luname,* command, keyword, and value to be authorized. See “Command identifiers” on page 234 for information on specifying command identifiers.

Examples of generic characters in PERMIT and PROTECT statements: The following examples assume you are using the NVSS command authorization table statements to define command authorization and that your NVSS domain name (*luname*) is MSO01. To authorize only NETOP1 to issue the GETCONID command with the AUTH keyword and a value of MASTER, include the following statements:

```
PROTECT *.MSO01.GETCONID.AUTH.MASTER
PERMIT NETOP1 *.MSO01.GETCONID.AUTH.MASTER
```

- To protect all other keywords on the GETCONID command, include the following statement:

  ```
  PROTECT *.MSO01.GETCONID.*
  ```

- It is not required to use a generic character in the value position, but since all of the GETCONID keywords have corresponding values, a command identifier of `*.MSO01.GETCONID.*.*` would be functionally equivalent.

- Some commands have keywords that are issued without a corresponding value. For example, the SAVE and PPT keywords of the EVERY command do not have a value. To authorize NETOP1 in domain MSO01 to issue the EVERY command with both the SAVE and PPT keywords, include the following statements:

  ```
  * PROTECT KEYWORDS ON "EVERY" COMMAND
  PROTECT *.MSO01.EVERY.SAVE
  PROTECT *.MSO01.EVERY.PPT
  PERMIT NETOP1 *.MSO01.EVERY.SAVE
  PERMIT NETOP1 *.MSO01.EVERY.PPT
  ```

- To protect both the SAVE and PPT keywords and all other keywords on the EVERY command, include the following statement:

  ```
  PROTECT *.MSO01.EVERY.*
  ```

- Notice that there is no generic character used for value. The command identifier `*.MSO01.EVERY.*.*` would not protect the SAVE and PPT keywords, but would only protect keywords that are specified with a corresponding value. The command identifier `*.MSO01.EVERY.*` protects keywords that have corresponding values as well as keywords that do not have corresponding values.
- Using an asterisk (*) as a trailing generic character at the end of a command identifier allows you match on subsequent values in that field and subsequent fields. Using a trailing asterisk in the *command* field will protect the command, and all its keywords and values. For example, if you use this statement:

  ```
  PROTECT *.*.STOP*
  ```

  It will protect the NVSS STOP command, and all its keywords and values. Note that this is equivalent to coding all three of the following statements:

  ```
  PROTECT *.*.STOP
  PROTECT *.*.STOP.*
  PROTECT *.*.STOP*.*
  ```

- Using a trailing asterisk in the *keyword* field will protect the keyword for that command, with all the values on that keyword. For example, to protect all the values on all the REXX keywords for the NVSS DEFAULTS command, use this statement:

  ```
  PROTECT *.*.DEFAULTS.REXX*
  ```

SETVAR statement: The SETVAR statement defines a table variable to represent multiple values which can be used in command identifiers. Table variables must represent an entire field value and must be defined before being used.

The syntax for the SETVAR statement is:

SETVAR

```
                             ┌─,────┐
                             ▼      │
►►──SETVAR variable_name ──┴─value─┴────────────────────────────────────────►◄
```

Where:

variable_name
: Is the 1–32 character name of the variable you are defining. The variable name cannot contain an ampersand (&), dash (-), period (.), asterisk (*), or percent sign (%).

value
: Is the 1–242 character value to be included in the command identifier. The value cannot contain an ampersand (&), dash (-), or period (.).

Examples:

To define a variable EURODOM to represent domains MSO01, CNM02, and CNM99, use the following:

```
SETVAR EURODOM MSO01,CNM02,CNM99
```

To subsequently use the variable &EURODOM in a PROTECT statement, include the following:

```
PROTECT *.&EURODOM.STOP
```

When processed, this generates the equivalent of the following table statements:

```
PROTECT *.MSO01.STOP
PROTECT *.CNM02.STOP
PROTECT *.CNM99.STOP
```

Note that the table variable EURODOM represented the entire field value. A specification such as the following is *not* valid:

```
SETVAR EURODOM 01,02,99
PROTECT *.CNM&EURODOM.STOP
```

To define a variable XDOM to represent commands ROUTE and RMTCMD, use the following:

```
SETVAR XDOM ROUTE,DSIUSNDM
```

To subsequently use the variable &XDOM in a PROTECT statement, include the following:

```
PROTECT *.MSO01.&XDOM
```

This generates the equivalent of the following table statements:

```
PROTECT *.MSO01.ROUTE
PROTECT *.MSO01.DSIUSNDM
```

Loading the NVSS command authorization table: The NVSS command authorization table can be loaded during NVSS initialization as specified by the OPTIONS statement in the DSIDMN member of DSIPARM. During initialization, if syntax errors are encountered, messages are issued but any valid statements in the table are still loaded. After NVSS initialization is complete, errors can be corrected and the table reloaded using the REFRESH command. If there are syntax errors in the table processed by the REFRESH command, the table is not loaded. There is a TEST keyword on the REFRESH command that you can use to check for syntax errors before attempting to load the table.

Using the NVSS command authorization table to protect VTAM command keywords and values: You can restrict any keywords and values of a VTAM command using the NVSS command authorization table. For values entered with the VTAM keyword ID, SLU, PLU, LU1, and LU2, if the VTAM resource is qualified with a network ID, access to the network ID and resource name are checked separately. So they should be defined in separate PROTECT statements. The VTAM resource name and the network ID can be up to 8 characters long. If IDTYPE=IPADDR is entered with the VTAM DISPLAY command, the value entered with the ID keyword is an IP address and can be longer than 8 characters.

For example, the following statements are defined in the NVSS command authorization table:

```
PROTECT *.*.DISPLAY.ID.NETA
PROTECT *.*.DISPLAY.ID.DSICRTR
PROTECT *.*.DISPLAY.ID.87/123/136/121
```

An NVSS operator would not be able to execute the following VTAM commands:

```
D NET,ID=NETA.NTVB5LU
D NET,ID=NETA.DSICRTR
D NET,ID=NETB.DSICRTR
D NET,ID=87.123.136.121,IDTYPE=IPADDR
```

The operator can, however, execute the following commands:

```
D NET,ID=NTVB5LUC
D NET,ID=NETB.NTVB5LUC
```

Command authorization table – usage notes: Some command identifiers are more specific than others. For example, the following table statements are ordered from most specific to least specific, as you can determine by comparing the character strings from left to right:

```
 PROTECT *.MSO01.STOP.FORCE.MSO01PPT
 PROTECT *.MSO01.STOP.FORCE.*
 PROTECT *.MSO01.STOP.*
```

The most specific PROTECT statement in your NVSS command authorization table is the statement with the generic character latest in the sequence of fields, after the *netid* and *luname* fields. Only the most specific statement that matches the command being issued is used for command authorization.

The type of generic character is also used to determine which command identifier is most specific. Because the percent sign (%) generic character replaces just a single character, the percent sign is considered more specific than the asterisk (*) generic character. For example, ABC% is more specific than ABC* when evaluating the value ABCD.

For example, the value SYS1 matches both the SY%1 and the SYS* identifiers. In this case, SYS* is considered to be more specific because the generic character is in the fourth position, rather than SY%1 which has a generic character in the third position.

If both a PROTECT and an EXEMPT statement are coded for the same command identifier, message BNH184E will be issued indicating a syntax error in the NVSS command authorization table.

If this message is issued due to a REFRESH command, the NVSS command authorization table is not loaded. If the message is issued during initialization, the NVSS command authorization table is loaded, but only the first (PROTECT or EXEMPT) statement is used. Use message BNH184E to find the problem.

Command authorization table example: The following steps provide an example of defining operator authority using an NVSS command authorization table:

1. Define groups of operators.

   ```
   GROUP GRP1 NETOP1,NETOP2,AUTO1,AUTO2
   <BEGIN>
   GROUP GRP2 OPER1,OPER2,OPER3,OPER4,OPER5,OPER6,NETOP1,NETOP2,
   AUTO1,AUTO2
   <END>
   ```

   Note that these operators have been grouped into two classes of authorization.
2. Define the commands, keywords, and values to be protected.

- The following statements define the OVERRIDE command as unprotected except for the REXXSTRF keyword. This keyword can only be used by operators in group GRP1.

```
EXEMPT       NETA.MS001.OVERRIDE
PROTECT      NETA.MS001.OVERRIDE.REXXSTRF.*
PERMIT GRP1  NETA.MS001.OVERRIDE.REXXSTRF.*
```

- All of the following statements are comments. If you remove the asterisks from these statements, they protect the GLOBALV command and restrict its use to operators in groups GRP1 and GRP2. The statements also protect the SAVEC and RESTOREC keywords, and restrict their use to operators in groups GRP1 and GRP2. Finally, the statements protect the asterisk (*) and PURGEC keywords, and restrict their use to operators in group GRP1.

```
* PROTECT      NETA.MS001.GLOBALV
* PERMIT GRP1  NETA.MS001.GLOBALV
* PERMIT GRP2  NETA.MS001.GLOBALV
* PROTECT      NETA.MS001.GLOBALV.SAVEC
* PERMIT GRP1  NETA.MS001.GLOBALV.SAVEC
* PERMIT GRP2  NETA.MS001.GLOBALV.SAVEC
* PROTECT      NETA.MS001.GLOBALV.RESTOREC
* PERMIT GRP1  NETA.MS001.GLOBALV.RESTOREC
* PERMIT GRP2  NETA.MS001.GLOBALV.RESTOREC
* PROTECT      NETA.MS001.GLOBALV.ASTERISK
* PERMIT GRP1  NETA.MS001.GLOBALV.ASTERISK
* PROTECT      NETA.MS001.GLOBALV.PURGEC
* PERMIT GRP1  NETA.MS001.GLOBALV.PURGEC
```

Determining the user identity used for authority checking commands

Authority checking restricts the ability of an operator or a task to use commands, keywords, and values.

Table 14 identifies the operator or task identifier that is used to authority check msys for Operations commands based on the command and environment. The identity is referred to as the ***SOURCEID***.

Table 14. SOURCEID determination

Command and environment	SOURCEID determination
EXCMD command or a same-domain LABEL command prefix used to queue an imbedded command to another task.	The SOURCEID is the task that issued the EXCMD command, or the existing SOURCEID at the time the EXCMD command was issued.
TIMER commands that are scheduled to run under the PPT.	The SOURCEID is the task that issued the AT, EVERY, CHRON, or AFTER command, or the existing SOURCEID at the time the AT, EVERY, CHRON, or AFTER command was issued. **Note:** The SOURCEID is not destroyed by saving and restoring timer commands.
NVSS SUBMIT command for jobs submitted to the operating system from NVSS.	If OPERSEC=SAFCHECK or OPERSEC=SAFDEF, the identity that is checked by the operating system is the issuer of the SUBMIT command, or the existing SOURCEID at the time the SUBMIT command was issued. For other values of OPERSEC, NVSS's authority is used for submitting the job.

Table 14. SOURCEID determination (continued)

Command and environment	SOURCEID determination
NVSS commands that were entered at an MVS operator console.	When an MVS console has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, NVSS commands can be entered from that MVS console. This is done by prefixing the NVSS command with the NVSS designator character, which by default is %. If the MVS operator has logged on to the MVS console with a user ID, the SOURCEID is the user ID of the MVS operator. If an operator has not logged on at the EMCS console, the SOURCEID of that task defaults to *BYPASS*. Commands issued from a source ID of *BYPASS* are not checked for command authorization by: • The NVSS command authorization table • The SAF product OPERCMDS class • The SAF product NETCMDS class **Note:** If a command is entered from the MVS master console, it will be routed to one of the following: • The autotask with the specific console name • The autotask with console name "*MASTER*" • The autotask with console name "*ANY*"
NVSS commands that are entered using the MVS MODIFY command.	When an MVS console has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, NVSS commands can be entered from that MVS console by issuing an MVS MODIFY or STOP command against the NVSS task. The NVSS command is entered as text following the MODIFY command. The first parameter on the MODIFY command is the application ID that is being modified. If the MVS operator has logged on to the MVS console with a user ID, the SOURCEID is the user ID of the MVS operator. If an operator has not logged on at the EMCS console, the SOURCEID of that task defaults to *BYPASS*. Commands issued from a source ID of *BYPASS* are not checked for command authorization by: • The NVSS command authorization table • The SAF product OPERCMDS class • The SAF product NETCMDS class **Note:** If a command is entered from the MVS master console, it will be routed to: • The autotask with the specific console name • The autotask with console name "*MASTER*" • The autotask with console name "*ANY*"
NVSS commands that were entered by TSO users.	When a TSO user ID has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, NVSS commands can be entered from that TSO user ID when the user is acting as an MVS operator by using an EMCS console session, or when using SDSF. The SOURCEID is the TSO user's user ID.
Commands issued from JCL.	When a job that issues a NVSS command is submitted by a TSO user ID, the SOURCEID is the TSO user ID. If the ID of the submitter is unknown, a default user ID is inserted. The value of the default user ID is defined by the system installation.
MVS ROUTE command issued from NVSS.	If the MVS command ROUTE is issued from a NVSS task, the originating source ID is always passed to the SAF product for authorization checks in the OPERCMDS class. This occurs for all settings of AUTHCHK and CMDAUTH.

Table 14. SOURCEID determination (continued)

Command and environment	SOURCEID determination
Commands that are routed to an operator from the automation table.	The SOURCEID is the operator ID to which the command is routed. **Note:** Commands from the automation table are subject to authority checking unless SEC=BY was specified on the CMDMDL statement or SEC=DE was specified (or SEC was not specified) and AUTOSEC=BYPASS is in effect. For more information, refer to the DEFAULTS command in the NVSS online help.
CNMSMSG service (PL/I and C).	If CNMSMSG is called to queue a command from one task to another, the SOURCEID is the task name (TVBOPID) of the CNMSMSG issuer, or the existing SOURCEID at the time the CNMSMSG service was called.

Understanding security for specific commands

This section provides additional information about protecting the following commands:

- CHRON
- EXCMD
- MVS
- SUBMIT

Defining security for the CHRON command: The CHRON command has syntax that is more complex than most commands. CHRON uses multiple levels of keywords, items in lists, and quoted strings.

Command security for the CHRON command is checked so that operands within parentheses can be uniquely defined in the command authorization table (CMDAUTH=TABLE).

The following rules describe CHRON commands and which command identifiers are checked:

Rule 1: Each keyword that does not take a value (NOSAVE, SAVE, LOCAL, GMT, REFRESH, TEST, and DEBUG) is checked in the form:

Command example:

```
netid.luname.CHRON.keyword
```

Rule 2: Each keyword with a value is checked in the form:

Command example:

```
netid.luname.CHRON.keyword.value
```

With the CHRON command, the value may be a list or quoted string.

Command example:

```
CHRON AT=(),RECOVERY=IGNORE,NOSAVE,LOCAL,ROUTE=OPER1,ID=TEST1,
COMMAND='MSG ALL HELLO'
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.AT.()
netid.luname.CHRON.RECOVERY.IGNORE
netid.luname.CHRON.NOSAVE
netid.luname.CHRON.LOCAL
netid.luname.CHRON.ROUTE.OPER1
netid.luname.CHRON.ID.TEST1
netid.luname.CHRON.COMMAND.'MSG_ALL_HELLO'
```

Rule 3A: Keywords appearing within parenthesized lists of other keywords are checked using the hierarchy of keywords with a ″(″ between so that the keyword hierarchy can be uniquely identified. The compound keyword that is generated is tested with the value of the innermost keyword. This checking is done at each level of the nesting of the lists. When a keyword is within a list that is the value of another keyword, the notation uses both keywords with a ″(″ between them.

Rule 3B: From the outermost to innermost, if a ″keyword=(list)″ appears, if any values appear in the list without keywords, the ″keyword=value″ check is done for that value. The keyword that is checked is the keyword hierarchy defined by Rule 3A.

Command example:

```
CHRON EVERY=(INTERVAL=(000-01.00.00 FOR=08.00.00))
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.EVERY.(INTERVAL=(000_01/00/00_FOR=08/00/00))
netid.luname.CHRON.EVERY(INTERVAL.(000_01/00/00_FOR=08/00/00)
netid.luname.CHRON.EVERY(INTERVAL.000_01/00/00
netid.luname.CHRON.EVERY(INTERVAL(FOR.08/00/00
```

Substitution of certain special characters is performed as described in “Protecting commands containing special characters” on page 82. For example, a dash becomes an underscore in the command identifier.

Rule 4: Quoted string values are checked as a single value, including the apostrophes and all text within the apostrophes.

Command example:

```
netid.luname.CHRON.REM.'ISN''T THIS A REMARK STRING?'
```

The following command identifier is checked:

```
CHRON REM='ISN''T THIS A REMARK STRING?'
```

Rule 5: For the DAYSWEEK keyword, days of the week can be followed by a sublist identifying particular weeks of the month. The day name and each item in the sublist are treated as a unit.

Command example:

```
CHRON EVERY=(DAYSWEEK=(NOT MON(1ST 2nd)))
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.EVERY.(DAYSWEEK=(NOT_MON(1ST_2ND)))
netid.luname.CHRON.EVERY(DAYSWEEK.(NOT_MON(1ST_2ND))
netid.luname.CHRON.EVERY(DAYSWEEK.NOT
netid.luname.EVERY(DAYSWEEK.MON(1ST)
netid.luname.EVERY(DAYSWEEK.MON(2ND)
```

This lets you check the sublist values without concern for the order of the items within the sublist. Notice that the value ″MON(1st 2nd)″ is not checked since the values MON(1st) and MON(2nd) are checked.

Table 15 illustrates a detailed list of possible command identifiers that may be defined for the CHRON command. The rule that causes the command identifier to be checked is shown in the second column.

Table 15. NVSS command identifiers for the CHRON command

Commands and keywords identifier	RULE	Command authorization table identifier
CHRON	Command name	netid.luname.CHRON
AT=		
	2	netid.luname.CHRON.AT.()
	2	netid.luname.CHRON.AT.(timespec datespec) [2]
	3B	netid.luname.CHRON.AT.timespec
	3B	netid.luname.CHRON.AT.datespec[2]
	2	netid.luname.CHRON.AT.yyy_mm_dd_hh/mm/ss/micros[2]
AFTER=	2	netid.luname.CHRON.AFTER.timespec [2]
	2	netid.luname.CHRON.AFTER.ddd_hh/mm/ss/micros [2]
EVERY=	2	netid.luname.CHRON.EVERY.NONE
	2	netid.luname.CHRON.EVERY.( )
	2	netid.luname.CHRON.EVERY.(everyoptions) [2]
EVERY=(INTERVAL=	3A	netid.luname.CHRON.EVERY(INTERVAL.( )
	3B	netid.luname.CHRON.EVERY(INTERVAL.(intervaloptions) [2]
	3B	netid.luname.CHRON.EVERY(INTERVAL.timespec [2]
	3A	netid.luname.CHRON.EVERY(INTERVAL. ddd_hh/mm/ss/micros [2]
EVERY=(INTERVAL= (FOR=	3A	netid.luname.CHRON.EVERY(INTERVAL(FOR.timespec
	3A	netid.luname.CHRON.EVERY(INTERVAL(FOR. hh/mm/ss/micros [2]
EVERY=(INTERVAL= (MXREPEAT=	3A	netid.luname.CHRON.EVERY(INTERVAL(MXREPEAT. NOLIMIT
	3A	netid.luname.CHRON.EVERY(INTERVAL(MXREPEAT. repeat_count
EVERY=(INTERVAL= (OFF=	3A	netid.luname.CHRON.EVERY(INTERVAL(OFF.timespec
	3A	netid.luname.CHRON.EVERY(INTERVAL(OFF. hh/mm/ss/micros [2]
EVERY=(REMOVE=	3A	netid.luname.CHRON.EVERY(REMOVE.MANUALLY
	3A, 3B	netid.luname.CHRON.EVERY(REMOVE.(removeoptions) [2]
	3B	netid.luname.CHRON.EVERY(REMOVE.datespec [2]
	3B	netid.luname.CHRON.EVERY(REMOVE.timespec [2]
	3A	netid.luname.CHRON.EVERY(REMOVE. yyyy_mm_dd_hh/mm/ss/micros [2]
EVERY= (REMAFTER=	3A	netid.luname.CHRON.EVERY(REMAFTER.timespec [2]
	3A	netid.luname.CHRON.EVERY(REMAFTER. ddd_hh/mm/ss/micros [2]

Table 15. NVSS command identifiers for the CHRON command (continued)

Commands and keywords identifier	RULE	Command authorization table identifier
EVERY= (DAYSWEEK=	3A 3B 3B 3B 5	netid.luname.CHRON.EVERY(DAYSWEEK.ALL netid.luname.CHRON.EVERY(DAYSWEEK.(daysweeklist) [2] netid.luname.CHRON.EVERY(DAYSWEEK.NOT netid.luname.CHRON.EVERY(DAYSWEEK.dayname netid.luname.CHRON.EVERY(DAYSWEEK. dayname (sublist_element) [2]
EVERY=(DAYSMON=	3A 3B 3B 3B	netid.luname.CHRON.EVERY(DAYSMON.ALL netid.luname.CHRON.EVERY(DAYSMON.(dayslist) [2] netid.luname.CHRON.EVERY(DAYSMON.NOT netid.luname.CHRON.EVERY(DAYSMON.dayofmonth [2]
EVERY=(CALENDAR=	3A 3B 3B 3B	netid.luname.CHRON.EVERY(CALENDAR.ALL netid.luname.CHRON.EVERY(CALENDAR.(calendarlist) [2] netid.luname.CHRON.EVERY(CALENDAR.NOT netid.luname.CHRON.EVERY(CALENDAR.keyname [2]
RECOVERY=	2 2 2	netid.luname.CHRON.RECOVERY.IGNORE netid.luname.CHRON.RECOVERY.AUTOLGN netid.luname.CHRON.RECOVERY.PURGE
SAVE	1	netid.luname.CHRON.SAVE
NOSAVE	1	netid.luname.CHRON.NOSAVE
LOCAL	1	netid.luname.CHRON.LOCAL
ID=	2	netid.luname.CHRON.ID.idname
NOTIFY=	2	netid.luname.CHRON.NOTIFY.(notifylists)
NOTIFY=(PURGE=	3B 3B	netid.luname.CHRON.NOTIFY(PURGE.(purgelist) netid.luname.CHRON.NOTIFY(PURGE.taskname
NOTIFY=(REMOVE=	3B 3B	netid.luname.CHRON.NOTIFY(REMOVE.(removelist) netid.luname.CHRON.NOTIFY(REMOVE.taskname
NOTIFY=(IGNORE=	3B 3B	netid.luname.CHRON.NOTIFY(IGNORE.(ignorelist) netid.luname.CHRON.NOTIFY(IGNORE.taskname
NOTIFY=(RUN=	3B 3B	netid.luname.CHRON.NOTIFY(RUN.(runlist) netid.luname.CHRON.NOTIFY(RUN.taskname
REFRESH	1	netid.luname.CHRON.REFRESH
TEST	1	netid.luname.CHRON.TEST
DEBUG	1	netid.luname.CHRON.DEBUG
COMMAND=	4	netid.luname.CHRON.COMMAND.'quoted string' [2]
REM=	4	netid.luname.REM.'quoted string' [3]

Defining EXCMD command authorization: The NVSS EXCMD command is used to send commands to another task.

There are two operands that are used when issuing the EXCMD command. One is the *operator_id* where the command is being sent, and the other is the *command* being sent. These two operands are checked as a keyword-value pair.

Note: When protecting the target verb of EXCMD, specify the command verb, not any synonym. Unless otherwise documented, the verb is the label used on the CMDMDL statement. The verb for labeled commands beginning with a slash is EXCMD

For example, the command identifier to protect `EXCMD OPER1 LOGOFF` is:

```
PROTECT *.*.EXCMD.OPER1.LOGOFF
```

Defining additional MVS command authority: You can protect individual MVS system commands from unauthorized use with the OPERCMDS class of an SAF product, such as RACF. This is additional authorization checking done at the MVS level, after the command security checking done by the NVSS command authorization table.

To protect MVS commands:

1. Ensure your OPERSEC setting has a value of SAFCHECK or SAFDEF.
2. Define command profiles to restrict specific commands from operators. For example, to restrict all operators from being able to issue an MVS QUIESCE command, enter:
   ```
   RDEFINE OPERCMDS MVS.QUIESCE UACC(NONE)
   ```
3. Ensure that the OPERCMDS class is active and enabled for processing. The following RACF commands can be used to do this:
   ```
   SETROPTS CLASSACT(OPERCMDS)
   SETROPTS RACLIST(OPERCMDS)
   ```
4. When the OPERCMDS class is active, use the RACF REFRESH function when you change a definition:
   ```
   SETROPTS RACLIST(OPERCMDS) REFRESH
   ```

Defining SUBMIT command authorization: You can protect jobs submitted from NVSS using the SUBMIT command. When the NVSS SUBMIT command is issued, the SUBMIT command can be protected using NVSS command authorization. By protecting at this level, you can stop the processing for unauthorized users before the job is ever submitted to the system.

Controlling access to data sets and members

To prevent unauthorized alteration of data, you can protect data sets with an SAF product, such as RACF. To prevent unauthorized viewing of passwords and other restricted information, protect them with NVSS commands such as READSEC and WRITESEC.

[3] This value may have a special character, such as "." or "-", for example in the programmer time notation. You substitute the character "/" for "." and "_" for "-" when making the security definition.

Data set security

You can restrict unauthorized alteration of data sets from the NVSS environment using the DATASET class of the security product. The following are some considerations when using the DATASET class of the security product:

- NVSS requires CONTROL access to the DSILOG data set to write to the netlog.
- NVSS requires READ access to the first data set identified by the DSILIST DD statement.
- NVSS requires READ access to non-DSIPARM data sets that are specified on the NVSS SUBMIT command.
- Each of the following NVSS commands require UPDATE access to the first data set identified by the DSILIST DD statement.
  - AUTOTBL (with the LISTING keyword)
  - AUTOCNT (with the FILE keyword)
  - QRYGLOBL (with the FILE keyword)
  - SECMIGR (with output to DSILIST)
- If you use SECMIGR to convert from the NVSS command authorization table to RACF (TBL2RACF), the operator running SECMIGR requires READ authority to the data set containing the NVSS command authorization table being converted.
- If you specify your own output data set, the operator running SECMIGR requires UPDATE authority to your output data set.

Note: NVSS trace records are not made for calls to the DATASET class, because the calls are made by MVS for the NVSS tasks.

Restricting access to data sets

To activate the data set protection described in the preceding section, do the following:

1. To enable task-level authorization checking, initialize NVSS product using OPTIONS values of OPERSEC=SAFCHECK or OPERSEC=SAFDEF. If you did not initialize the NVSS product using these values, you can also change the OPERSEC values using the NVSS REFRESH command.
2. If you are using an SAF product, add profiles for the data sets you want to protect. The RACF product requires that the highest-level qualifier of the data set name be either a task or group name.

   For example, use the RACF ADDSD command to add data set profiles. From an authorized TSO user, enter the following command to protect the OPER1.STATS data set:

   ```
   ADDSD 'OPER1.STATS'
   ```
3. If you are using an SAF product, authorize the operator tasks so they can access the data set. For example, use the RACF PERMIT command to authorize operator tasks to the data set. To authorize NETOP1 to have update access to OPER1.STATS, enter the following command from an authorized TSO user:

   ```
    PERMIT 'OPER1.STATS' CLASS(DATASET) ID(NETOP1) ACCESS(UPDATE)
   ```

NVSS READSEC and WRITESEC commands

Use the NVSS READSEC and WRITESEC commands to restrict access to data sets and members by NVSS commands. When you specify security for the READSEC command, it affects all of the NVSS commands which can display sensitive information, such as:

- BROWSE with a member name
- NCCF LIST with the CLIST or PROFILE keywords
- PIPE stages

  - < (From disk)
  - QSAM
- VSAM command DSIVSMX

Using READSEC and WRITESEC is the only way to prevent operators from viewing data sets and members using these NVSS commands. In NVSS, security is defined so that operators have access to DSIOPEN and msys for Operations online help. DSIOPEN is a DD name designed to hold information which should not be secured, such as NEWS data and PF key definitions. Anything other than DSIOPEN and the online help may be considered sensitive information.

Because attempts to define security for these NVSS commands is considered a severe error, message BNH115A is generated every time an operator logs on. The error text for this message is “SPECIAL SECURITY IN EFFECT FOR BROWSE AND READSEC”, which indicates NVSS has defined default protection for sensitive data sets and members, and the NVSS commands which display data sets or members will fail. You must delete any security definitions for the commands and reinitialize NVSS to clear the error condition.

If you use command authorization without specifying values for READSEC and WRITESEC, operators will have access to all data sets and members.

Do not protect DD name CNMPNL1, operators need to access online help that is contained there.

For more information about how to use the READSEC and WRITESEC commands, refer to the online help.

Appendix B. Return codes

This appendix provides information about macro return codes and about return codes from VIEW and BROWSE (see "VIEW and BROWSE return codes" on page 262).

Macro return codes

This section provides return codes for the following macros:

- DSICES
- DSIDKS, on page 256
- DSIMQS; on page 258
- DSIPRS, on page 258
- DSIPSS, on page 259
- DSIPUSH, on page 260
- DSIZCSMS, on page 261
- DSIZVSMS, on page 261

Macro DSICES return codes in Register 15

The following return codes for macro DSICES are found in register 15:

0 The function is successful. One of the following describes what occurred:

- A regular command is found in the system command table and the address of the SCT entry is returned.
- The verb is not found in the SCT (if you specify CLISTCK, a command list is found with the specified name), and the dummy SCT entry for a command list is returned.

4 The command that is found can be processed as a regular or immediate command; the address is returned.

8 An immediate command is found in the system command table; the address is returned.

12 The module is not found, or there is an incorrect verb length; no address is returned.

16 The operator is not authorized to issue the command. This is caused by the security definitions that are in place for this command. No address is returned in SCTADDR.

This return code is not applicable if MODNAME was specified.

20 Either the command found is incompatible with the task type that called the routine, and the address is returned; or you specified CLISTCK=YES and the request is issued in an asynchronous exit, and the address is not returned.

24 You specified CLISTCK=YES but the command or command list is not found in DSISCT or DSICLD.

28 You specified CLISTCK=YES but storage requested for CLISTCK processing is not obtained.

32 NVSS internal error.

36 An unexpected return code was received from the security authorization facility (SAF). Message BNH238E is issued with the SAF return code inserted. This return code is not applicable if MODNAME was specified as this specification causes no authorization checking to be performed.

40 Authorization to the command is not granted because the security environment for the operator cannot be established. Message BNH239E is issued when this condition is first encountered to provide the security product return code information. Message BNH273I is issued when the condition has been corrected. This return code is not applicable if MODNAME was specified as this specification causes no authorization checking to be performed.

44 Authorization to the command is not granted because an unexpected return code was received from the command authorization table. Message BNH199E is issued indicating the command identifier and the operator ID being checked. This return code is not applicable if MODNAME was specified as this specification causes no authorization checking to be performed.

48 Authorization to the command is not granted because the NVSS internal security information containing the source ID of the command could not be found. Message BNH277E is issued identifying the command being checked. This return code is not applicable if MODNAME was specified as this specification causes no authorization checking to be performed.

52 Authorization to the command is not granted because the source ID is blank in the NVSS internal security information. Message BNH277E is issued identifying the command being checked. This return code is not applicable if MODNAME was specified because this specification causes no authorization checking to be performed.

Macro DSIDKS return codes in Register 15

The return codes and code meanings in register 15 are dependent on the “TYPE=” specification.

The following return codes are for TYPE=CONN:

0 The function is successful. Data control blocks and I/O buffer are obtained and initialized.

4 An incorrect data set name.

12 No storage was available for I/O buffer.

The following return codes are for TYPE=FIND:

0 The function is successful. The member or file is found and the first record is read.

4 The member or file is not found in the source statement library or in the specified library, or an empty member or file is found.

8 The member or file is found but an I/O error occurred on the first read.

12 The specified definition name or data set has not been opened.

20 The specified control block identifier is not valid; the member or file is not found.

28 There is a syntax error in the %INCLUDE card.

36 There is an incorrect member name on the %INCLUDE card.

40 There is an incorrect embed member, which can cause a deadlock condition. (This occurs when a member embeds itself.)

44 An unrecoverable system error occurred. An internal NVSS service failed, because of a storage failure.

46 An I/O error is encountered while trying to include a member specified in a %INCLUDE statement.

100 + *xx*
An error occurred during CLOSE processing. msys for Operations attempts to recover the data set after a failure during a previous FIND or READ. Refer to the description of the *xx* return code under the CLOSE macro in your operating system macro reference.

200 + *xx*
An error occurred during OPEN processing. msys for Operations attempts to recover the data set after a failure during a previous FIND or READ. Refer to the description of the *xx* return code under the OPEN macro in your operating system macro reference.

The following return codes are for TYPE=DISC:

0 The disconnect is successful; data and I/O buffers are freed successfully.

20 The specified control block identifier is not valid and no storage is freed.

46 An I/O error is encountered while trying to INCLUDE a member specified in a %INCLUDE statement.

The following return codes are for TYPE=READ:

0 The function is successful; the record is read.

4 The end of data is reached.

8 An I/O error occurred during reading.

12 Reading of this record is prohibited; an I/O error may have occurred, the end of data may have been reached, or the caller did not issue TYPE=FIND first.

20 The specified control block identifier is not valid; the record is not read.

28 There is a syntax error in the %INCLUDE card.

36 A member name on the %INCLUDE card is not valid.

40 An embed member is not valid and can cause a deadlock condition. (This occurs when a member embeds itself.)

44 An unrecoverable system error occurred. An internal service failed, because of a storage failure.

46 An I/O error is encountered while trying to include a member specified in a %INCLUDE statement.

100 + *xx*
An error occurred during CLOSE processing. msys for Operations attempts to recover the data set after a failure during a previous FIND or READ. Refer to the description of the *xx* return code under the CLOSE macro in your operating system macro reference.

200 + *xx*
An error occurred during OPEN processing. msys for Operations attempts to recover the data set after a failure during a previous FIND or READ. Refer to the description of the *xx* return code under the OPEN macro in your operating system macro reference.

Macro DSIMQS return codes in Register 15

The following return codes for macro DSIMQS are found in register 15:

0 The function was successful; the message is queued.

4 The buffer length was either:
- Not greater than 0
- Less than the combined length of HDRBLEN plus HDRTDISP
- Greater than 32000

8 The operator ID designated as the receiver of authorized messages was not found.

12 A buffer could not be obtained or dynamic resource control failed.

16 msys for Operations is terminating; the external request cannot be completed.

20 The SWB address is not valid.

22 The list specified with the LIST option contained no operator IDs. It contained only unassigned group IDs.

23 Messages were routed to the first 255 operators or groups, or both.

24 The value specified for priority was not valid.

26 The internal function request for the command to be run contains the IFRAUTBC or IFRAUTBN fields. The task that receives this command has no MQS receipt support and cannot process these fields.

28 A message stack enquiry failed.

32 NVSS internal error.

Note: When a command procedure written in REXX or NVSS command list language is executing, NVSS services all message queues, except the low-priority queue, at three points:
- Initially, before the first instruction
- After the execution of any NVSS command
- Throughout the period of any wait state (for &WAIT, &PAUSE, or WAIT)

Because of this, two command lists queued at the same time to the high- or normal-priority queues appear to run in reverse order. The first one is initiated, then before it executes its first instruction, it is preempted, and the second command executes. To have command lists execute in the order queued, always queue them at low priority.

Macro DSIPRS return codes in Register 15

The following return codes for macro DSIPRS are found in register 15:

0 The function was successful. The required size of the table was returned in PDBSIZE, or the command was parsed and the parse table was built.

4 The input buffer was parsed, but there was no data in the input buffer (0 length data) or the data in the input buffer was all blanks. Only the buffer address and number of entries (0) could be returned in the parse table.

8 The parse table was too small for the input buffer; a partial parse table was built, and the number of entries was set to the number that the parse table could hold. The size of the parse table should be increased.

12 Unbalanced quotes. Returned only if SUB=YES is specified.

16 The number of characters between two consecutive delimiters in the input buffer was greater than 255.

20 An unpaired double-byte character set (DBCS) delimiter of DBCS data bytes was found in the input buffer. For example, one of the following may have occurred:

- The end of the input buffer was found before the DBCS data-ending delimiter shift-in (X'0F').
- A second DBCS data-beginning delimiter shift-out (X'0E') was found before the DBCS data-ending delimiter shift-in (X'0F').
- An odd number of DBCS data bytes were found between DBCS data delimiters.

100 No PDB or an incorrect PDB was passed; or no PDBSIZE or an incorrect PDBSIZE was passed.

Note: You must specify the operands DELIM, FIRST, and SUB, identically, in the pair of DSIPRS parse commands issued. Otherwise, the second parse can fail or the storage can be overlaid.

Macro DSIPSS return codes in Register 15

The following return codes for macro DSIPSS are found in register 15:

0 The function was successful; the message is written. For TYPE=PSSWAIT, an ECB has been posted. Check the ECB list to determine which event has completed. For TYPE=ASYPANEL, the send or receive request has passed NVSS syntax and buffer checking and has been sent to VTAM; it does not indicate the success or failure of VTAM completion of the receive. You must check the ECB post code to determine the success or failure of the ASYPANEL request. The post code is put into the ECB specified in the panel parameter list.

4 For TYPE=XSEND, no request parameter list (RPL) was found and no data was sent.

8 Parameter error. There is an error in the formatting of the message buffer header. For TYPE=XSEND, the session is not active and no data is sent. For TYPE=ASYPANEL, the parameter list is inconsistent. If you specify the output buffer, you must also specify the length. If you specify the input ECB, you must specify the input area address, input area length, and the data length address of the returned length.

12 There is not enough storage available to complete the request. No output is sent, and the input command processor cannot be scheduled.

16 DSIPSS TYPE=OUTPUT was issued for an immediate command or in an IRB exit routine. Use DSIPSS TYPE=IMMED or DSIMQS instead.

20 No terminal session exists. For TYPE=ASYPANEL, the panel request came from a task other than an OST. No input is received. For TYPE=CANCEL, the panel request came from a task other than an OST.

36 For TYPE=ASYPANEL, a temporary error occurred. The contents of the panel have been modified. Reformat the panel using an Erase/Write or Erase/Write Alternate 3270 command. Then retry the request.

40 A permanent I/O error occurred. Do not retry the request. No output is sent, and no input processor is scheduled. For TYPE=ASYPANEL, no input is received. For TYPE=CANCEL, NVSS is unable to restart normal terminal activity.

48 For TYPE=ASYPANEL, no I/O is scheduled because the command processor issued a second DSIPSS TYPE=ASYPANEL requesting input before the previous request had completed.

56 For TYPE=PSSWAIT or TYPE=TESTWAIT, at least one NVSS ECB was posted.

68 For TYPE=OUTPUT or TYPE=IMMED, a message being processed for the RMTCMD command failed to be transmitted. This error can occur when the DSIUDST task is inactive.

The following ECB post codes for PSS TYPE=ASYPANEL are found in the ECB if you specified one:

0 The function was successful; the requested data is available.

12 There is not have enough storage available to complete the request. The output data was sent, but the input data is not available.

36 A temporary error occurred during a full-screen read. Retry the request. The output data was sent, but the input data is not available.

40 A permanent error occurred during a full-screen read. Do not retry the request. The output data was sent, but the input data is not available.

52 The requested input was canceled by DSIPSS TYPE=CANCEL. Do not retry the request immediately. The output data was sent, but the input data is not available.

Macro DSIPUSH return codes in Register 15

The following return codes for macro DSIPUSH are found in register 15:

0 The function was successful; the long-running command request is queued.

4 Storage is not available for the request.

8 The ABEND reinstate or LOGOFF routine is required but was not specified.

12 The request was issued from an incorrect task:
- RESUME request issued under DST
- ABEND request issued under DST
- DSIPUSH issued and task is not an OST, NNT, DST, or PPT

16 The request was issued while in an immediate command or while the NVSS program is in an exit, or in the middle of a LOGOFF routine or ABEND reinstate routine.

20 The RESUME routine is a command list, or the CMDMDL statement did not pass validity checking, or the operator's scope class does not permit access to the RESUME routine.

Verify that the first CMDMDL statement for this command in DSICMD is not type immediate.

24 The ABEND reinstate routine is a command list, or the CMDMDL statement did not pass validity checking, or the operator's scope class does not permit access to the RESUME routine.

Verify that the first CMDMDL statement for this command in DSICMD is not type immediate.

28 The LOGOFF routine is a command list, or the CMDMDL statement did not pass validity checking.

Verify that the first CMDMDL statement for this command in DSICMD is not type immediate.

32 The macro invocation is not valid. Fix assembly errors before trying to run the program.

Macro DSIZCSMS return codes in Register 15

The following major return codes for macro DSIZCSMS are found in register 15:

0 The function was successful; data was sent to VTAM.

4 The requested function could not be performed.

8 The input buffer was too small to build a forward RU.

12 An error was found in a parameter specification.

16 The program did not execute under a data services task.

20 The RULENG exceeded the maximum RU length required.

Macro DSIZCSMS minor return codes in Register 0

The following minor return codes for macro DSIZCSMS are found in register zero (0):

0 The function was successful.

4 The SWB was not valid.

8 The DSRB was not valid.

12 The DSRB that was passed was in use.

16 An unsolicited DSRB was passed.

20 An operator ID specified in the DSRB was not valid.

24 Reserved.

28 There was insufficient storage to process the request.

32 The CNMI is inactive.

36 The request was rejected by the access method.

48 The specified SECONDS value was not valid.

Macro DSIZVSMS return codes in Register 15

The following major return codes for macro DSIZVSMS are found in register 15:

0 Successful completion of VSAM function.

4 Manipulative macro error occurred during processing.

8 An error occurred in the EXECUTE form of a manipulative macro. An operand was not in the list.

12 Unsuccessful completion.

16 DSIZVSMS was issued while not executing under a DST.

Macro DSIZVSMS minor return codes in Register 0

The following minor return codes for macro DSIZVSMS are found in register zero (0):

0 Successful completion.

4 The specified DSRB was not valid or in use.

8 An ACB was unavailable or was not open. This may be due to a `SWITCH` *`taskname`*`,T` command having been issued.

12 Resume verb processing error.

16 An installation exit rejected the request.

20 The VSAM I/O request was not valid or there was an I/O scheduling error.

24 Data truncation occurred during substitution of data in an installation exit; or control block storage could not be obtained.

28 An installation exit returned a return code that was not valid.

Note: For more information about specifying FUNC and OPTION, refer to the OS/VS VSAM library. For an explanation of RPL feedback codes, refer to the OS/VS VSAM library and the MVS/ESA™ library.

VIEW and BROWSE return codes

Table 16 lists and describes the return codes that can be received for the VIEW and BROWSE command. The table also provides a brief description of the action you need to take.

Table 16. Return codes from VIEW and BROWSE

Code	Meaning	Your action
4	• Specified panel not found in CNMPNL1, CNMMSGF, or CNMCMDF data sets (MVS). • Possible input/output (I/O) error.	Put panel definition in correct data set or file.
8	Panel definition format not valid; no noncomment lines found.	Correct format of panel definition.
12	You are not authorized to browse the member.	Ask your system programmer to redefine your authorization.
16	VIEW command processor invoked with parameters that are not valid. *Name1* must be 1 to 8 characters and *name2* must be a valid panel ID. Valid parameters are INPUT, NOINPUT, MSG, NOMSG.	Correct command list to use valid option.
24	Full-screen command processor is available to OST only.	Do not invoke VIEW from a non-OST.
28	Change file to logical record length of 80 bytes.	
32	Unrecoverable error resulted from macro call. Error could be that CNMMSGF or CNMCMDF has not been installed for online message or command help. Also, refer to message DWO050I in the NVSS log.	Install CNMMSGF or CNMCMDF. Otherwise, call IBM for service.
36	Unrecoverable internal programming error occurred. Also, refer to message DWO050I in the NVSS log.	Call IBM for service.

Table 16. Return codes from VIEW and BROWSE (continued)

Code	Meaning	Your action
40	Browse panel CNMBROWS, which is used for browsing members, was not found.	Put CNMBROWS in correct data set or file.
81	Panel definition format not valid; no text indicator line found, or more than 49 option definitions found.	Correct format of panel definition.
83	Panel definition format not valid; comment lines in wrong place.	Correct format of panel definition.

Appendix C. Coexistence of msys for Operations and SA OS/390 releases

This appendix gives you information about ensuring compatibility of msys for Operations with the following SA OS/390 releases:

- SA OS/390 V1R3
- SA OS/390 V2R1
- SA OS/390 V2R2

SA OS/390 V1R3

When you migrate your system to OS/390 R10 or later, and you are using SA OS/390 V1R3, you can run msys for Operations and SA OS/390 V1R3 on the same systems if the following requirements are met:

- msys for Operations and SA OS/390 V1R3 must belong to different XCF groups. Since the XCF group suffix for msys for Operations should not be changed, this implies that the suffix for SA OS/390 V1R3 must not be A0.
- To ensure that msys for Operations and SA OS/390 V1R3 do not generate the same console names (for the requirement of unique console names see “EMCS console names and the automation router task” on page 34) you must change the global variable AOFCNMASK in AOFEXDEF (see *System Automation for OS/390 Customization Version 1 Release 3*, Appendix F). For example, you could specify `AOFCNMASK=290C0D0E0F101`**`5`**`18`. The default is 290C0D0E0F101718.
- Certain functions of SA OS/390 V1R3 that are also available in msys for Operations must be switched off in SA OS/390 V1R3. These are:
  - – WTO buffer recovery

    To switch off WTO buffer recovery perform the following steps:

    1. Select the **MVS Component** policy object.
    2. Select the WTOBUF AUTOMATION policy of **MVS Component**.
    3. Set the **Recovery** flag to NO.
  - – The CFDRAIN command

    This command is available in msys for Operations as the DRAIN subcommand of the INGCF command. To disable the CFDRAIN command in SA OS/390 V1R3 remove the CFDRAIN synonym from AOFCMD.

SA OS/390 V2R1

SA OS/390 V2R1 is fully compatible with msys for Operations.

SA OS/390 V2R2

SA OS/390 V2R2 is fully compatible with msys for Operations.

Appendix D. msys for Operations customization checklist

The following summarizes the customization steps necessary to implement Managed System Infrastructure for Operations (msys for Operations), including SPE UW99415. These activities need to be performed on each system following the installation. If the same PROCLIB and PARMLIB data sets are shared among the participating systems, then these changes need only be made once on any system and will be applicable to every system.

This appendix provides examples that you can use to quickly set up msys for Operations. Once you have become familiar with the various functions, you can then make your own changes.

Step 1: Creating shared and system-unique VSAM and non-VSAM data sets

Start by creating the shared and system unique VSAM and non-VSAM data sets. The shared DSIPARM data set will contain customized, installation unique msys for Operations settings. We recommend that you share this data set because this will minimize setup activities. The sample job to perform this activity is member INGALLC0 in ING.SINGSAMP. Modify this job so that a set of unique data sets is created for *each participating system*, as shown in Figure 58 on page 268. For more information, see “Allocating data sets and VSAM clusters using job INGALLC0 and INGALLC4” on page 39.

Note: The 2nd job step copies a seed record into the IPLDATA cluster after it is created. This record is in stream under data definition name (ddname) LOWKEY and the 1st 20 bytes *must be hexadecimal zeros* and not spaces or blanks (X'40's) as they would appear:

```
                    10Y
00000000000000000000FFE4444444444444444444444444444444444444444444444444
00000000000000000000108000000000000000000000000000000000000000000000000000
```

```
//RONNMSOA JOB (034D000,TS),NORTHRUP,CLASS=C,MSGCLASS=T,
//             NOTIFY=RONN
//MSOALOC PROC Q1=MSOPS,              ** DSN HIGH LEVEL QUALIFIER
//             DOMAIN=,               ** MSYS/OPS DOMAIN NAME
//             SER=TOTSTJ,            ** WHERE TO ALLOCATE DATA SET
//             DSN=,                  ** Low LVL DSN QUALIFIER
//             PS=,                   ** PRIMARY PDS SPACE
//             RECFM=FB,              ** RECORDFORMAT
//             LRECL=80,              ** RECORDLENGTH
//             BLKSIZE=6160,          ** BLOCKSIZE
//             DIR=                   ** DIRECTORY BLOCKS FOR PDS SPACE
//STEP1   EXEC PGM=IEFBR14
//DSNMODEL DD  DSN=&amp&Q1  &DOMAIN..&DSN,
//             DISP=(NEW,CATLG,CATLG),
//             SPACE=(CYL,(&PS,1,&DIR)),
//             DCB=(RECFM=&RECFM,LRECL=&LRECL,BLKSIZE=&BLKSIZE),
//             UNIT=SYSDA,VOL=SER=&SER//        PEND
//DSIPARM EXEC MSOALOC,DSN=DSIPARM,PS=5,DIR=30,DOMAIN=SHARED,BLKSIZE=3920
//DSILST1 EXEC MSOALOC,DOMAIN=MSO47,DSN=DSILIST,PS=1,DIR=10
//DSILST2 EXEC MSOALOC,DOMAIN=MSO54,DSN=DSILIST,PS=1,DIR=10
//DSILST3 EXEC MSOALOC,DOMAIN=MSO55,DSN=DSILIST,PS=1,DIR=10
//*
//STEP2   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DATA,DLM='||'
 /* *************************************************************** */
 /* Define Shared VSAM Cluster For Saving IPL Related Information   */
 /* *************************************************************** */
 SET MAXCC=0
 DEF CLUSTER(NAME(MSOPS.SHARED.IPLDATA)                   -
          INDEXED                                         -
          NOREUSE                                         -
          IMBED                                           -
          KEYS(20 0)                                      -
          RECSZ(90 4089)                                  -
          CISZ(4096)                                      -
          SHR(4 3)                                        -
          REC(10000 1000))                                -
          VOL(TOTSTJ)                                     -
        DATA(NAME(MSOPS.SHARED.IPLDATA.DATA))             -
        INDEX(NAME(MSOPS.SHARED.IPLDATA.INDEX))
 IF LASTCC = 0 THEN                                       -
   DO
   REPRO IFILE(LOWKEY)                                    -
           ODS(MSOPS.SHARED.IPLDATA)
   END
||
//LOWKEY   DD  DATA,DLM='||'
                    10Y
||
//*                   +-- Y/N SAVE PARMLIB DATA W/ OR W/O COMMENTS
//*                 ++--- NN  NUMBER OF IPL RECORDS PER SYSTEM
//*
//STEP3   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DATA,DLM='||'
```

Figure 58. Combination of sample job members INGALLC0 and INGALLC4 in ING.SINGSAMP (Part 1 of 6)

```
/* *************************************************************** */
/* Define System SC47 Unique VSAM Clusters - msys/Ops Domain MSO47 */
/* *************************************************************** */
DEF CLUSTER(NAME(MSOPS.MSO47.DSILOGP)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(125,404)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(4096))                      -
       INDEX                              -
        (CISZ(1024)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO47.DSILOGS)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(125,404)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(4096))                      -
       INDEX                              -
        (CISZ(1024)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO47.DSITRCP)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(114,146)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(16384))                     -
       INDEX                              -
        (CISZ(512)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO47.DSITRCS)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(114,146)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(16384))                     -
       INDEX                              -
        (CISZ(512)                        -
         IMBED)
```

Figure 58. Combination of sample job members INGALLC0 and INGALLC4 in ING.SINGSAMP (Part 2 of 6)

```
DEF CLUSTER(NAME(MSOPS.MSO47.DSISVRT)     -
         INDEXED                          -
         SHR(2)                           -
         VOL(TOTSTJ)                      -
         CYLINDERS(2 1)                   -
         KEYS(54 0)                       -
         RECSZ(64 0512)                   -
         FSPC(5 5)                        -
         REUSE)                           -
        DATA                              -
        (CISZ(8192))                      -
       INDEX                              -
        (CISZ(4096)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO47.STATS)       -
         VOL(TOTSTJ)                      -
         KEYS(20 0)                       -
         RECSZ(252 252)                   -
         FSPC(0 0)                        -
         SHR(2)                           -
         CISZ(512)                        -
         INDEXED                          -
         REUSE                            -
         IMBED)                           -
       DATA                               -
        (NAME(MSOPS.MSO47.STATS.DATA)     -
         CYL(2 0))                        -
       INDEX                              -
        (NAME(MSOPS.MSO47.STATS.INDEX)    -
         TRK(2 0))
/* *************************************************************** */
/* Define System SC54 Unique VSAM Clusters - msys/Ops Domain MSO54 */
/* *************************************************************** */
DEF CLUSTER(NAME(MSOPS.MSO54.DSILOGP)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(125,404)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(4096))                      -
       INDEX                              -
        (CISZ(1024)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.DSILOGS)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(125,404)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(4096))                      -
       INDEX                              -
        (CISZ(1024)                       -
         IMBED)
```

Figure 58. Combination of sample job members INGALLC0 and INGALLC4 in ING.SINGSAMP (Part 3 of 6)

```
DEF CLUSTER(NAME(MSOPS.MSO54.DSITRCP)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(114,146)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(16384))                     -
       INDEX                              -
        (CISZ(512)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.DSITRCS)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(114,146)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(16384))                     -
       INDEX                              -
        (CISZ(512)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.DSISVRT)     -
         INDEXED                          -
         SHR(2)                           -
         VOL(TOTSTJ)                      -
         CYLINDERS(2 1)                   -
         KEYS(54 0)                       -
         RECSZ(64 0512)                   -
         FSPC(5 5)                        -
         REUSE)                           -
       DATA                               -
        (CISZ(8192))                      -
       INDEX                              -
        (CISZ(4096)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.STATS)       -
         VOL(TOTSTJ)                      -
         KEYS(20 0)                       -
         RECSZ(252 252)                   -
         FSPC(0 0)                        -
         SHR(2)                           -
         CISZ(512)                        -
         INDEXED                          -
         REUSE                            -
         IMBED)                           -
       DATA                               -
        (NAME(MSOPS.MSO54.STATS.DATA)     -
         CYL(2 0))                        -
       INDEX                              -
        (NAME(MSOPS.MSO54.STATS.INDEX)    -
         TRK(2 0))
```

Figure 58. Combination of sample job members INGALLC0 and INGALLC4 in ING.SINGSAMP (Part 4 of 6)

```
/* *************************************************************** */
/* Define System SC55 Unique VSAM Clusters - msys/Ops Domain MSO55 */
/* *************************************************************** */
DEF CLUSTER(NAME(MSOPS.MSO55.DSILOGP)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(125,404)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(4096))                      -
       INDEX                              -
        (CISZ(1024)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO55.DSILOGS)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(125,404)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(4096))                      -
       INDEX                              -
        (CISZ(1024)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO55.DSITRCP)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(114,146)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(16384))                     -
       INDEX                              -
        (CISZ(512)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO55.DSITRCS)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(114,146)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(16384))                     -
       INDEX                              -
        (CISZ(512)                        -
         IMBED)
```

Figure 58. Combination of sample job members INGALLC0 and INGALLC4 in ING.SINGSAMP (Part 5 of 6)

```
 DEF CLUSTER(NAME(MSOPS.MSO55.DSISVRT)       -
          INDEXED                            -
          SHR(2)                             -
          VOL(TOTSTJ)                        -
          CYLINDERS(2 1)                     -
          KEYS(54 0)                         -
          RECSZ(64 0512)                     -
          FSPC(5 5)                          -
          REUSE)                             -
        DATA                                 -
         (CISZ(8192))                        -
        INDEX                                -
         (CISZ(4096)                         -
          IMBED)
 DEF CLUSTER(NAME(MSOPS.MSO55.STATS)         -
          VOL(TOTSTJ)                        -
          KEYS(20 0)                         -
          RECSZ(252 252)                     -
          FSPC(0 0)                          -
          SHR(2)                             -
          CISZ(512)                          -
          INDEXED                            -
          REUSE                              -
          IMBED)                             -
        DATA                                 -
         (NAME(MSOPS.MSO55.STATS.DATA)       -
          CYL(2 0))                          -
        INDEX                                -
         (NAME(MSOPS.MSO55.STATS.INDEX)      -
          TRK(2 0))
||
//
```

Figure 58. Combination of sample job members INGALLC0 and INGALLC4 in ING.SINGSAMP (Part 6 of 6)

Step 2: Adding additional procedures to PROCLIB

Five additional procedures need to be copied into a PROCLIB data set, usually SYS1.PROCLIB, pointed to by the PROC00 DD definition statement in the JES2 procedure. The sample members are INGNVAP0, HSAPIPLC, INGPHOM, INGPIPLC and INGPIXCU and can be found in ING.SINGSAMP. For more information, see “Adding Procedures to PROCLIB” on page 41.

The 1st procedure, INGNVAP0, starts msys for Operations and requires installation-specific customization as highlighted in Figure 59 on page 274. It may be renamed to anything you choose.

```
//MSOPS   PROC DOMAIN=MSO&SYSCLONE., ** MSYS/OPS DOMAIN NAME
//             SQ1='NETVIEW',        ** NVSS DSN HLQ
//             SQ3='ING',            ** MSAS DSN HLQ
//             VQ1=MSOPS            ** VSAM DSN HLQ
//MSOPS   EXEC PGM=DSIMNT,TIME=1440,REGION=64M,DPRTY=(13,13),
//             PARM=(24K,200,'&DOMAIN','','','','')
//SYSPRINT DD  SYSOUT=*
//STEPLIB  DD  DSN=&SQ3..SINGMOD1,DISP=SHR
//         DD  DSN=&SQ1..CNMLINK,DISP=SHR
//DSICLD   DD  DSN=&SQ3..SINGNREX,DISP=SHR
//         DD  DSN=&SQ1..CNMCLST,DISP=SHR
//         DD  DSN=&SQ1..CNMSAMP,DISP=SHR
//DSIOPEN  DD  DSN=&SQ1..SDSIOPEN,DISP=SHR
//DSIPARM  DD  DSN=MSOPS.SHARED.DSIPARM,DISP=SHR
//         DD  DSN=&SQ3..SINGNPRM,DISP=SHR
//         DD  DSN=&SQ1..DSIPARM,DISP=SHR
//DSILIST  DD  DSN=MSOPS.&DOMAIN..DSILIST,DISP=SHR
//DSIVTAM  DD  DSN=ESA.SYS1.VTAMLST,DISP=SHR
//DSIPRF   DD  DSN=&SQ3..SINGNPRF,DISP=SHR
//         DD  DSN=&SQ1..DSIPRF,DISP=SHR
//DSIMSG   DD  DSN=&SQ3..SINGNMSG,DISP=SHR
//         DD  DSN=&SQ1..SDSIMSG1,DISP=SHR
//BNJPNL1  DD  DSN=&SQ1..BNJPNL1,DISP=SHR
//BNJPNL2  DD  DSN=&SQ1..BNJPNL2,DISP=SHR
//CNMPNL1  DD  DSN=&SQ3..SINGNPNL,DISP=SHR
//         DD  DSN=&SQ1..CNMPNL1,DISP=SHR
//         DD  DSN=&SQ1..SEKGPNL1,DISP=SHR
//DSILOGP  DD  DSN=&VQ1..&DOMAIN..DSILOGP,
//             DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSILOGS  DD  DSN=&VQ1..&DOMAIN..DSILOGS,
//             DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSITRCP  DD  DSN=&VQ1..&DOMAIN..DSITRCP,
//             DISP=SHR,AMP=AMORG
//DSITRCS  DD  DSN=&VQ1..&DOMAIN..DSITRCS,
//             DISP=SHR,AMP=AMORG
//DSISVRT  DD  DSN=&VQ1..&DOMAIN..DSISVRT,
//             DISP=SHR,AMP=AMORG
//AOFSTAT  DD  DSN=&VQ1..&DOMAIN..STATS,
//             DISP=SHR,AMP=AMORG
//HSAIPL   DD  DSN=&VQ1..SHARED.IPLDATA,
//             DISP=SHR,AMP=AMORG
```

Figure 59. Installation specific customization

The 2nd procedure, HSAPIPLC, should be run immediately after an IPL and causes information related to the IPL to be written away for later retrieval and comparison. Ensure that the correct high level qualifiers (HLQs) are used and add the entry `COM='S HSAPIPLC'` to COMMNDxx to automatically start this procedure following an IPL.

```
//HSAPIPLC PROC HLQ1=ING,HLQ2=MSOPS
//COLLECT  EXEC PGM=HSAPSIPL,REGION=2M
//STEPLIB  DD   DISP=SHR,DSN=&HLQ1..SINGMOD1
//HSAIPL   DD   DISP=SHR,DSN=&HLQ2..IPLDATA
```

The other procedures require no changes, are used dynamically for internal processes and should be copied as is. If not copied to SYS1.PROCLIB ensure that the Proclib data set is concatenated to the IEFPDSI data definition statement in SYS1.PARMLIB(MSTRJCL).

Step 3: Authorizing and linking data sets

From the following list of data sets, the first three need to be authorized. In addition, the SINGMOD2 data set must be added to the LNKLST concatenation and the SINGMOD3 data set to the LPALST concatenation. For more information, see "Updating the link list, LPA, and APF authorizations with PROG*xx*" on page 39.

1. ING.SINGMOD1
2. ING.SINGMOD2—Add to *LNKLST* Concatenation
3. NETVIEW.CNMLINK
4. ING.SINGMOD3—Add to *LPALST* Concatenation

To accomplish this, statements need to be added to the active *PROGxx* and *LPALSTxx* members of SYS1.PARMLIB which would take effect at the next IPL.

To make the changes permanent, add the following statements to the LNKLST and APF sections of the active PROG*xx* member. Add the single statement to the active LPALST*xx* member. Be careful to match the LNKLST set name (here LNKLST00) to the one actually in use and change the volume parameter to match the volser on which these data sets are allocated.

1. PROG*xx* LNKLST change:

   ```
   LNKLST ADD NAME(LNKLST00) DSN(ING.SINGMOD2) VOLUME(TOTSTJ)
   ```

2. PROG*xx* APF changes:

   ```
   APF ADD DSNAME(ING.SINGMOD1)          VOLUME(TOTSTJ)
   APF ADD DSNAME(ING.SINGMOD2)          VOLUME(TOTSTJ)
   APF ADD DSNAME(NETVIEW.CNMLINK)       VOLUME(TOTSTJ)
   ```

3. LPALST*xx* change:

   ```
   ING.SINGMOD3(TOTSTJ)
   ```

To make the changes dynamically, the most straightforward approach is to create a new member (PROGMO) based on the example member, INGPROG0 located in ING.SINGSAMP. The following statements show what is needed. Choose an appropriate LNKLST set name and change the volume parameter to match the volser on which these data sets are allocated. Be aware that dynamic changes to LPA requires the data set to be cataloged.

```
LNKLST DEFINE NAME(LNKLSTMO) COPYFROM(CURRENT)
LNKLST ADD NAME(LNKLSTMO) DSNAME(ING.SINGMOD2) VOLUME(TOTSTJ) ATTOP
LNKLST ACTIVATE NAME(LNKLSTMO)
APF ADD DSNAME(ING.SINGMOD1)                  VOLUME(TOTSTJ)
APF ADD DSNAME(ING.SINGMOD2)                  VOLUME(TOTSTJ)
APF ADD DSNAME(NETVIEW.CNMLINK)               VOLUME(TOTSTJ)

A                        LPA ADD MASK(*) DSNAME(ING.SINGMOD3)
```

Once PROGMO has been setup, issue the command `SET PROG=MO`. This command results in the merging of these changes with current settings. *Dynamic changes only remain effective until the next IPL.*

Step 4: Updating the active SCHEDxx member

A Program Properties Table entry needs to be added to the active *SCHED*xx member of SYS1.PARMLIB. The statements to perform this activity can be found in the sample INGSCHE0 in ING.SINGSAMP. For more information, see “Updating member SCHED*xx*” on page 38.

```
/*               NVSS DSIMNT                                 */
PPT PGMNAME(DSIMNT)               /* PROGRAM NAME NETVIEW      */
    KEY(8)                        /* PROTECTION KEY            */
    NOSWAP                        /* NON-SWAPPABLE             */
```

Step 5: Updating the active MPFLSTxx member

The active Message Processing Facility List, *MPFLST*xx in SYS1.PARMLIB needs to be updated (for more information, see “Updating member MPFLST*xx*” on page 38). If no form of automation is in use, then the statements to perform this activity can be found in sample member INGEMPF in ING.SINGSAMP. The only statements required are:

```
.NO_ENTRY,SUP(NO),RETAIN(I,CE),AUTO(YES)
.DEFAULT,SUP(YES),RETAIN(I,CE),AUTO(NO)
```

However, if the installation’s current processing is reliant on existing MPFLSTxx settings which may be incompatible with the above, then the following specific statements will need to be worked into the active member instead:

```
AOF603D,SUP(NO),RETAIN(NO),AUTO(YES)
AOF*,SUP(NO),RETAIN(NO),AUTO(NO)
IEA230E,SUP(NO),RETAIN(NO),AUTO(YES)
IEA231A,SUP(NO),RETAIN(NO),AUTO(YES)
IEA232I,SUP(NO),RETAIN(NO),AUTO(YES)
IEA404A,SUP(NO),RETAIN(NO),AUTO(YES)
IEA405E,SUP(NO),RETAIN(NO),AUTO(YES)
IEA406I,SUP(NO),RETAIN(NO),AUTO(YES)
IEA794I,SUP(NO),RETAIN(NO),AUTO(YES)
IEA*,SUP(NO),RETAIN(NO),AUTO(NO)
IEE037D,SUP(NO),RETAIN(NO),AUTO(YES)
IEE041I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE043I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE205I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE400I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE503I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE533E,SUP(NO),RETAIN(NO),AUTO(YES)
IEE600I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE712I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE769E,SUP(NO),RETAIN(NO),AUTO(YES)
IEE889I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE*,SUP(NO),RETAIN(NO),AUTO(NO)
ILR009E,SUP(NO),RETAIN(NO),AUTO(YES)
ILR*,SUP(NO),RETAIN(NO),AUTO(NO)
INGY1097I,SUP(NO),RETAIN(NO),AUTO(YES)
INGY*,SUP(NO),RETAIN(NO),AUTO(NO)
IRA200E,SUP(NO),RETAIN(NO),AUTO(YES)
IRA201E,SUP(NO),RETAIN(NO),AUTO(YES)
IRA202I,SUP(NO),RETAIN(NO),AUTO(YES)
IRA204E,SUP(NO),RETAIN(NO),AUTO(YES)
IRA*,SUP(NO),RETAIN(NO),AUTO(NO)
IXC102A,SUP(NO),RETAIN(NO),AUTO(YES)
IXC247D,SUP(NO),RETAIN(NO),AUTO(YES)
IXC250I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC251I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC255I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC263I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC309I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC402D,SUP(NO),RETAIN(NO),AUTO(YES)
IXC500I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC501A,SUP(NO),RETAIN(NO),AUTO(YES)
IXC517I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC559I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC560A,SUP(NO),RETAIN(NO),AUTO(YES)
IXC*,SUP(NO),RETAIN(NO),AUTO(NO)
IXG257I,SUP(NO),RETAIN(NO),AUTO(YES)
IXG261E,SUP(NO),RETAIN(NO),AUTO(YES)
IXG*,SUP(NO),RETAIN(NO),AUTO(NO)
IXL126I,SUP(NO),RETAIN(NO),AUTO(YES)
IXL127A,SUP(NO),RETAIN(NO),AUTO(YES)
IXL*,SUP(NO),RETAIN(NO),AUTO(NO)
```

VTAM Requirements

msys for Operations requires VTAM to be operational on every participating system in the sysplex (for more information, see “Preparing VTAM” on page 43).

Fortunately, Cross-System Coupling Facility (XCF) services are most often used for VTAM-to-VTAM communication within a sysplex virtually eliminating the need for any VTAM definitions. Essentially, the first VTAM to start creates an XCF group called ISTXCF. This forms the basis of communication with other sysplex VTAMs. As subsequent VTAMs initialize, they join the same XCF group and are dynamically recognized.

There are two considerations:

1. **XCF signaling** must be configured to use a Coupling Facility, Channel-to-Channel Connections or both.
2. **XCFINIT=NO** *must not* be specified in the ATCSTRxx member that is used during VTAM initialization.

Step 6: Defining the application major nodes to VTAM

Define the msys for Operations application major nodes to VTAM (for more information, see "Preparing VTAM" on page 43). The statements to perform this activity can be found in the sample INGVTAM member in ING.SINGSAMP. This will be a new member which can be named anything you choose, such as APMSO*xx*. It must be placed in the data set pointed to by the VTAMLST data definition statement in the VTAM procedure. This data set must also be coded on the DSIVTAM data definition statement of the procedure used to start msys for Operations as described in Step 2: Adding additional procedures to PROCLIB. Once this member is in place, the current ATCCON*xx* member should be edited to include the new member name just created. This will ensure that VTAM activates these APPLs during startup.

```
******************************************************************************
**    THIS APPLICATION MAJNODE DEFINES msys/Ops (NVSS) TO VTAM              **
**    DOMAIN: MSO&SYSCLONE.                                                 **
******************************************************************************
          VBUILD TYPE=APPL
******************************************************************************
* MSYS_OPS MAIN TASK                                                         *
******************************************************************************
MSO&SYSCLONE.  APPL AUTH=(VPACE,ACQ,PASS),PRTCT=MSO&SYSCLONE.,                 X
               MODETAB=AMODETAB,DLOGMOD=DSIL6MOD,                              X
               APPC=YES,PARSESS=YES,                                           X
               DMINWNL=4,DMINWNR=4,DSESLIM=8,VPACING=10,                       X
               AUTOSES=2
*****************************************************************************
* MSYS_OPS PRIMARY POI - (PROGRAM OPERATOR INTERFACE)                       *
*****************************************************************************
MSO&SYSCLONE.PPT APPL AUTH=(NVPACE,SPO),PRTCT=MSO&SYSCLONE.,EAS=1,             X
               MODETAB=AMODETAB,DLOGMOD=DSILGMOD
******************************************************************************
* MSYS_OPS SUBTASKS                                                          *
******************************************************************************
MSO&SYSCLONE.* APPL AUTH=(NVPACE,SPO,ACQ),PRTCT=MSO&SYSCLONE.,EAS=4,           X
               MODETAB=AMODETAB,DLOGMOD=DSILGMOD
```

Alternatively, activation can be performed dynamically using the following command:

```
'V NET,ACT,ID=APMSOXX'
```

Step 7: Customizing the security definitions

A Make the determined security definition changes. The statements to perform this
A activity can be found in the sample INGSAF1 in NETVIEW.DSIPARM. The
A recommendation is to use a SAF product, and the statements in this member
A assume that Security Server (RACF) is used. If this is not being used, the
A statements will need to be altered to conform with the product that is in use.

Step 7: Customizing the security definitions

A Extensive customization is necessary here as highlighted in the following example.
A The highlighted statements show which statements are necessary to add an
A additional operator called RONN and domains called MSO47, MSO54, and MSO55.
A For more information see Chapter 4, ″Making security definitions″. Chapter 4,
A “Making security definitions,” on page 75.

```
//RONNSAF JOB (POK,999),NORTHRUP,CLASS=C,MSGCLASS=T,NOTIFY=&SYSUID
//STP1 EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *,DLM=@@
/*
/***********************************************************************
/* To activate the classes needed for msys for Operations and protect
/* against unauthorized logon, change 'domain_name' to the domain
/* name specified in your msys for Operations startup procedure.
/***********************************************************************
SETROPTS CLASSACT(APPL)
SETROPTS CLASSACT(NETCMDS) GRPLIST
RDEF APPL MSO47 UACC(NONE)
RDEF APPL MSO54 UACC(NONE)
RDEF APPL MSO55 UACC(NONE)
/*
/***********************************************************************
/* To define the task identifiers needed for msys for Operations,
/* change 'domain_name' to your msys for Operations domain name and
/* 'SSIR_task_name' to the CNMCSSIR task name you specified in
/* CNMSTYLE.
/***********************************************************************
ADDUSER MSO47PPT
ADDUSER MSO54PPT
ADDUSER MSO55PPT
ADDUSER MSO47SIR
ADDUSER MSO54SIR
ADDUSER MSO55SIR
/*
```

Figure 60. Sample member CNMSAF1 in NETVIEW.DSIPARM (Part 1 of 8)

```
     /***********************************************************************
     /* msys for Operations autotasks - DO NOT CHANGE
     /***********************************************************************
     ADDUSER AUTO1
     ALTUSER AUTO1 NETVIEW(IC(LOGPROF2) MSGRECVR(NO))
     ADDUSER AUTO2
     ALTUSER AUTO2 NETVIEW(IC(LOGPROF4) MSGRECVR(NO))
     ADDUSER DBAUTO1
     ALTUSER DBAUTO1  NETVIEW(IC(LOGPROF4) MSGRECVR(NO))
     ADDUSER DSILCOPR
     ALTUSER DSILCOPR NETVIEW(MSGRECVR(NO))
     ADDUSER AUTBASE
     ALTUSER AUTBASE  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTGSS
     ALTUSER AUTGSS   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTMON
     ALTUSER AUTMON   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTMSG
     ALTUSER AUTMSG   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTNET1
     ALTUSER AUTNET1  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTREC
     ALTUSER AUTREC   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTSYS
     ALTUSER AUTSYS   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTLOG
     ALTUSER AUTLOG   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTCON
     ALTUSER AUTCON   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTRPC
     ALTUSER AUTRPC   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTWRK01
     ALTUSER AUTWRK01 NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTWRK02
     ALTUSER AUTWRK02 NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTWRK03
     ALTUSER AUTWRK03 NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTJES
     ALTUSER AUTJES   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTSHUT
     ALTUSER AUTSHUT  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTXCF
     ALTUSER AUTXCF   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
     ADDUSER AUTXCF2
     ALTUSER AUTXCF2  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
A    ADDUSER AUTHW001
A    ALTUSER AUTHW001 NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW002
A    ALTUSER AUTHW002 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW003
A    ALTUSER AUTHW003 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW004
A    ALTUSER AUTHW004 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW005
A    ALTUSER AUTHW005 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW006
A    ALTUSER AUTHW006 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW007
A    ALTUSER AUTHW007 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW008
A    ALTUSER AUTHW008 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW009
A    ALTUSER AUTHW009 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW010
A    ALTUSER AUTHW010 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW011
A    ALTUSER AUTHW011 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW012
A    ALTUSER AUTHW012 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW013
A    ALTUSER AUTHW013 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW014
A    ALTUSER AUTHW014 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW015
A    ALTUSER AUTHW015 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW016
A    ALTUSER AUTHW016 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW017
A    ALTUSER AUTHW017 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
A    ADDUSER AUTHW018
A    ALTUSER AUTHW018 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
```

```
/***********************************************************************
/* Edit the following defaults appropriately to define your operators.
/***********************************************************************
ADDUSER RONN
ALTUSER RONN     NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER1
ALTUSER OPER1    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER2
ALTUSER OPER2    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER3
ALTUSER OPER3    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER4
ALTUSER OPER4    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER5
ALTUSER OPER5    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER6
ALTUSER OPER6    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER NETOP1
ALTUSER NETOP1   NETVIEW(IC(LOGPROF1) MSGRECVR(YES))
ADDUSER NETOP2
ADDUSER OPER6
ALTUSER OPER6    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER NETOP1
ALTUSER NETOP1   NETVIEW(IC(LOGPROF1) MSGRECVR(YES))
ADDUSER NETOP2
ALTUSER NETOP2   NETVIEW(IC(LOGPROF1) MSGRECVR(YES))
/*
/**********************************************************************
/* Group your operators according to their responsibilities and
/* roles. Add as many CONNECT statements as you need. Each operator
/* can be connected to as many groups as needed.
/*
/* Operators are connected to a group according the functions they
/* are permitted run. They must also be connected to every lower group
/* as well. For example, Operators connected to MSYSOPS3 would also
/* require access to functions in the lower groups. To establish this
/* they must be connected to MSYSOPS1 and MSYSOPS2 in order to have
/* the appropriate access authority assigned to them.
/*
/* If users are not listed as member of any group, they will still
/* be able to use the INGPLEX and INGCF commands for display
/* purposes and any other msys for Operations commands that are
/* not specifically protected.
/*
```

Figure 60. Sample member CNMSAF1 in NETVIEW.DSIPARM (Part 3 of 8)

```
    /*********************************************************************
A   /* Users listed in this group are allowed to do most restricted base
A   /* NVSS commands. NOTE: The A/* CF command relies on EXCMD and
A   /* therefore requires the same authorization as EXCMD ie:
A   /* GROUP(MSYSOPS4)
    /*********************************************************************
    ADDGROUP MSYSOPS0
    CONNECT  NETOP1   GROUP(MSYSOPS0) UACC(READ)
    CONNECT  NETOP2   GROUP(MSYSOPS0) UACC(READ)
    CONNECT  OPER1    GROUP(MSYSOPS0) UACC(READ)
    CONNECT  OPER2    GROUP(MSYSOPS0) UACC(READ)
    CONNECT  OPER3    GROUP(MSYSOPS0) UACC(READ)
    CONNECT  OPER4    GROUP(MSYSOPS0) UACC(READ)
    CONNECT  OPER5    GROUP(MSYSOPS0) UACC(READ)
A   CONNECT  RONN     GROUP(MSYSOPS0) UACC(READ)
    /*********************************************************************
    /* Users listed in this group are allowed to execute FORCE and
    /* REBUILD actions on structures.
    /*********************************************************************
    ADDGROUP MSYSOPS1
    CONNECT  NETOP1   GROUP(MSYSOPS1) UACC(READ)
    CONNECT  NETOP2   GROUP(MSYSOPS1) UACC(READ)
    CONNECT  OPER1    GROUP(MSYSOPS1) UACC(READ)
A   CONNECT  RONN     GROUP(MSYSOPS1) UACC(READ)
    /*********************************************************************
    /* Users listed in this group are allowed to execute the SETXCF
    /* command with parm "ACOUPLE" and "PSWITCH".
    /*********************************************************************
    ADDGROUP MSYSOPS2
    CONNECT  NETOP1   GROUP(MSYSOPS2) UACC(READ)
    CONNECT  NETOP2   GROUP(MSYSOPS2) UACC(READ)
    CONNECT  OPER2    GROUP(MSYSOPS2) UACC(READ)
A   CONNECT  RONN     GROUP(MSYSOPS2) UACC(READ)
```

Figure 60. Sample member CNMSAF1 in NETVIEW.DSIPARM (Part 4 of 8)

```
   /********************************************************************
   /* Users listed in this group are allowed to execute the full
   /* functionality of INGCF ENABLE and INGCF DRAIN (without HW
   /* action ACTIVATE/DEACTIVATE).
   /********************************************************************
   ADDGROUP MSYSOPS3
   CONNECT  NETOP1   GROUP(MSYSOPS3) UACC(READ)
   CONNECT  NETOP2   GROUP(MSYSOPS3) UACC(READ)
   CONNECT  OPER3    GROUP(MSYSOPS3) UACC(READ)
A  CONNECT  RONN     GROUP(MSYSOPS3) UACC(READ)
   /********************************************************************
   /* Users listed in this group are allowed to do most restricted
   /* base NVSS commands and ACF COLD (due to EXCMD).
   /********************************************************************
   ADDGROUP MSYSOPS4
   CONNECT  NETOP1   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  NETOP2   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  OPER1    GROUP(MSYSOPS4) UACC(READ)
   CONNECT  OPER2    GROUP(MSYSOPS4) UACC(READ)
   CONNECT  OPER3    GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTO1    GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTO2    GROUP(MSYSOPS4) UACC(READ)
   CONNECT  DBAUTO1  GROUP(MSYSOPS4) UACC(READ)
   CONNECT  DSILCOPR GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTBASE  GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTGSS   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTMON   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTMSG   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTNET1  GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTREC   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTSYS   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTLOG   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTCON   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTRPC   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTWRK01 GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTWRK02 GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTWRK03 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW003 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW004 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW005 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW006 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW007 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW008 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW009 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW010 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW011 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW012 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW013 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW014 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW015 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW016 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW017 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW018 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW019 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW020 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW021 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW022 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW023 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW024 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW025 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW026 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW027 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW028 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW029 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW030 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW031 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW032 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTHW033 GROUP(MSYSOPS4) UACC(READ)
A  CONNECT AUTPLEX  GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTJES   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTSHUT  GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTXCF   GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTXCF2  GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTHW001 GROUP(MSYSOPS4) UACC(READ)
   CONNECT  AUTHW002 GROUP(MSYSOPS4) UACC(READ)
   CONNECT  RONN     GROUP(MSYSOPS4) UACC(READ)
```

Figure 60. Sample member CNMSAF1 in NETVIEW.DSIPARM (Part 5 of 8)

```
/*********************************************************************
A /* Users listed in this group are allowed the CLOSE and ACF COLD
A /* commands.
/*********************************************************************
ADDGROUP MSYSOPS5
CONNECT  NETOP1   GROUP(MSYSOPS5) UACC(READ)
CONNECT  NETOP2   GROUP(MSYSOPS5) UACC(READ)
CONNECT  RONN     GROUP(MSYSOPS5) UACC(READ)
/*
/*********************************************************************
/* To allow your operators to log on to msys for Operations, change
/* 'domain_name' to your domain name. Be sure to add PERMIT
/* statements for any operators not connected to any groups.
/* NOTE: Autotasks do not have to be permitted to log on.
/*********************************************************************
PERMIT MS047 CLASS(APPL) ID(MSYSOPS0) ACCESS(READ)
PERMIT MS047 CLASS(APPL) ID(MSYSOPS1) ACCESS(READ)
PERMIT MS047 CLASS(APPL) ID(MSYSOPS2) ACCESS(READ)
PERMIT MS047 CLASS(APPL) ID(MSYSOPS3) ACCESS(READ)
PERMIT MS047 CLASS(APPL) ID(MSYSOPS4) ACCESS(READ)
PERMIT MS047 CLASS(APPL) ID(MSYSOPS5) ACCESS(READ)
PERMIT MS054 CLASS(APPL) ID(MSYSOPS0) ACCESS(READ)
PERMIT MS054 CLASS(APPL) ID(MSYSOPS1) ACCESS(READ)
PERMIT MS054 CLASS(APPL) ID(MSYSOPS2) ACCESS(READ)
PERMIT MS054 CLASS(APPL) ID(MSYSOPS3) ACCESS(READ)
PERMIT MS054 CLASS(APPL) ID(MSYSOPS4) ACCESS(READ)
PERMIT MS054 CLASS(APPL) ID(MSYSOPS5) ACCESS(READ)
PERMIT MS055 CLASS(APPL) ID(MSYSOPS0) ACCESS(READ)
PERMIT MS055 CLASS(APPL) ID(MSYSOPS1) ACCESS(READ)
PERMIT MS055 CLASS(APPL) ID(MSYSOPS2) ACCESS(READ)
PERMIT MS055 CLASS(APPL) ID(MSYSOPS3) ACCESS(READ)
PERMIT MS055 CLASS(APPL) ID(MSYSOPS4) ACCESS(READ)
PERMIT MS055 CLASS(APPL) ID(MSYSOPS5) ACCESS(READ)
SETROPTS RACLIST(APPL) REFRESH
/*
/**********************************************************************
/*  Add FACILITY class resources for the Internal Hardware Transport &
/*  dynamic CDS functions.
/**********************************************************************
SETROPTS GENERIC(FACILITY)
RDEF FACILITY HSA.ET32*                UACC(NONE)
RDEF FACILITY MVSADMIN.LOGR            UACC(NONE)
RDEF FACILITY MVSADMIN.XCF.ARM         UACC(NONE)
RDEF FACILITY MVSADMIN.XCF.CFRM        UACC(NONE)
RDEF FACILITY MVSADMIN.XCF.SFM         UACC(NONE)
```

Figure 60. Sample member CNMSAF1 in NETVIEW.DSIPARM (Part 6 of 8)

```
/*********************************************************************
/* Add RDEF statements to the following list to define resources to
/* the NETCMDS class for any additional commands to which you wish to
/* restrict access.
/*********************************************************************
SETROPTS GENERIC(NETCMDS)
RDEF NETCMDS  *.*.*                     UACC(READ)
RDEF NETCMDS  *.*.ACF.COLD              UACC(NONE)
RDEF NETCMDS  *.*.INGAUTO               UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGCF.STR    UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGPLEX.CDS  UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGCF.CF     UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGPLEX.HW   UACC(NONE)
RDEF NETCMDS  *.*.REFRESH               UACC(NONE)
RDEF NETCMDS  *.*.DEFAULTS              UACC(NONE)
RDEF NETCMDS  *.*.LOGONPW               UACC(NONE)
RDEF NETCMDS  *.*.TS                    UACC(NONE)
RDEF NETCMDS  *.*.RID                   UACC(NONE)
RDEF NETCMDS  *.*.PURGE                 UACC(NONE)
RDEF NETCMDS  *.*.ALLOCATE              UACC(NONE)
RDEF NETCMDS  *.*.FREE                  UACC(NONE)
RDEF NETCMDS  *.*.EXCMD                 UACC(NONE)
RDEF NETCMDS  *.*.MODIFY                UACC(NONE)
RDEF NETCMDS  *.*.VARY                  UACC(NONE)
RDEF NETCMDS  *.*.CNME2008              UACC(NONE)
RDEF NETCMDS  *.*.MVS                   UACC(NONE)
RDEF NETCMDS  *.*.START                 UACC(NONE)
RDEF NETCMDS  *.*.STOP                  UACC(NONE)
RDEF NETCMDS  *.*.SWITCH                UACC(NONE)
RDEF NETCMDS  *.*.RESETDB               UACC(NONE)
RDEF NETCMDS  *.*.AUTOTBL               UACC(NONE)
RDEF NETCMDS  *.*.AUTOTASK              UACC(NONE)
RDEF NETCMDS  *.*.EZLEF002              UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXCPU       UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXCPU.*     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXIO        UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXIO.*      UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQIN      UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQIN.*    UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQOUT     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQOUT.*   UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXSTG       UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXSTG.*     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.SLOWSTG      UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.SLOWSTG.*    UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.REXXSTRF     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.REXXSTRF.*   UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.TASK         UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.TASK.*       UACC(NONE)
RDEF NETCMDS  *.*.CLOSE                 UACC(NONE)
/************* Protect access data sets *******************************
RDEF NETCMDS *.*.READSEC.DSIPARM.*     UACC(NONE)
RDEF NETCMDS *.*.WRITESEC.*            UACC(NONE)
RDEF NETCMDS *.*.WRITESEC.*.*          UACC(NONE)
/************* General "disallow" statements ***************************
RDEF NETCMDS *.*.CNME1087              UACC(NONE)
RDEF NETCMDS *.*.DSIZKNYJ              UACC(NONE)
RDEF NETCMDS *.*.DSIUSNDM              UACC(NONE)
RDEF NETCMDS *.*.FOCALPT               UACC(NONE)
RDEF NETCMDS *.*.NPDA                  UACC(NONE)
/*
```

Figure 60. Sample member CNMSAF1 in NETVIEW.DSIPARM (Part 7 of 8)

```
/***********************************************************************
/*  Add data set names with special authority requirements. Change
/*  'MSOPS' to the high level qualifier you have chosen.
/***********************************************************************
AD   MSOPS.**                           UACC(READ)
AD   MSOPS.*.DSILOGP                    UACC(READ)
AD   MSOPS.*.DSILOGS                    UACC(READ)
AD   MSOPS.*.DSILIST                    UACC(READ)
/*
/**********************************************************************
/* Add PERMIT statements to the following list to permit the
/* appropriate operators and groups to use the restricted commands
/* you added to the list above.
/**********************************************************************
PE *.*.INGAUTO              CLASS(NETCMDS)  ID(MSYSOPS0) ACCESS(READ)
PE *.*.INGRCCHK.INGCF.STR   CLASS(NETCMDS)  ID(MSYSOPS1) ACCESS(READ)
PE *.*.INGRCCHK.INGCF.CF    CLASS(NETCMDS)  ID(MSYSOPS3) ACCESS(READ)
PE *.*.INGRCCHK.INGPLEX.CDS CLASS(NETCMDS)  ID(MSYSOPS2) ACCESS(READ)
PE *.*.REFRESH              CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.DEFAULTS             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.LOGONPW              CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.TS                   CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.RID                  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.PURGE                CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.ALLOCATE             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.FREE                 CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.EXCMD                CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.MODIFY               CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.VARY                 CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.CNME2008             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.MVS                  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.START                CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.STOP                 CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.SWITCH               CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.RESETDB              CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.AUTOTBL              CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.AUTOTASK             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.EZLEF002             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXCPU      CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXCPU.*    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXIO       CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXIO.*     CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQIN     CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQIN.*   CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQOUT    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQOUT.*  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXSTG      CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXSTG.*    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.SLOWSTG     CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.SLOWSTG.*   CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.REXXSTRF    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.REXXSTRF.*  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.TASK        CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.TASK.*      CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.READSEC.DSIPARM.*    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.WRITESEC.*           CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.WRITESEC.*.*         CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.AOFRCFGA.COLD        CLASS(NETCMDS)  ID(MSYSOPS5) ACCESS(READ)
PE *.*.CLOSE                CLASS(NETCMDS)  ID(MSYSOPS5) ACCESS(READ)
PE *.*.INGRCCHK.INGPLEX.HW  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE HSA.ET32*                CLASS(FACILITY) ID(SYS1)     ACCESS(CONTROL)
PE HSA.ET32TGT.netid.*      CLASS(FACILITY) ID(SYS1)     ACCESS(CONTROL)
PE MSOPS.*.DSILOGP          CLASS(DATASET)  ID(MSYSOPS5) ACCESS(CONTROL)
PE MSOPS.*.DSILOGS          CLASS(DATASET)  ID(MSYSOPS5) ACCESS(CONTROL)
PE MSOPS.*.DSILIST          CLASS(DATASET)  ID(MSYSOPS5) ACCESS(ALTER)
PE MVSADMIN.XCF.ARM         CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
PE MVSADMIN.XCF.CFRM        CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
PE MVSADMIN.XCF.SFM         CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
PE MVSADMIN.LOGR            CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
/*
SETROPTS RACLIST(NETCMDS,FACILITY) REFRESH
@@
//
```

Figure 60. Sample member CNMSAF1 in NETVIEW.DSIPARM (Part 8 of 8)

There is an alternate, less robust method of defining msys for Operations userids without using an SAF product if you prefer. However, be aware that this is considerably less secure and that functions dependent on the Internal Hardware Transport will still require SAF authorization. In Step 1: Creating shared and system-unique VSAM and non-VSAM data sets, a shared user DSIPARM data set was created. This data set was placed ahead of the others in the DSIPARM data definition concatenation in the MSOPS procedure as shown in “Step 2: Adding additional procedures to PROCLIB” on page 273. Copy members DSIDMNK and DSIOPFU from NETVIEW.DSIPARM into this data set. Make changes to the options in these members as described in the “Step 8: Customizing the CNMSTYLE style sheet and members DSIDMNK, DSIOPFU and DSICMPRC” on page 288. If use of CF management capabilities (CF Drain and CF Enable) and partitioning of failed systems (IXC102A) is required then the following RACF definitions must be made.

```
//RONNSAF  JOB (POK,999),NORTHRUP,CLASS=C,MSGCLASS=T,NOTIFY=&SYSUID
//STP1    EXEC PGM=IKJEFT01
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *,DLM=@@
/*
/**********************************************************************
/*  MANDATORY RACF DEFINITIONS REQUIRED BY INTERNAL HARDWARE TRANSPORT
/**********************************************************************
/*  Define the default userid that will be associated with msys for
/*  Operations started tasks, the values 'STC', 'SYS1', 'SAFADMIN' and
/*  'MSO1234' may be changed. In most situations this type of userid
/*  will already be defined.
/**********************************************************************
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
ADDUSER  STC DFLTGRP(SYS1) OWNER(SAFADMIN) PASSWORD(MSO1234) OPERATIONS
SETROPTS RACLIST(STARTED) REFRESH
/*
/**********************************************************************
/*  msys for Operations Autotasks
/**********************************************************************
ADDUSER AUTHW001 NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW002 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW003 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW004 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW005 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW006 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW007 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW008 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW009 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW010 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW011 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW012 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW013 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW014 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW015 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW016 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW017 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW018 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW019 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW020 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW021 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW022 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW023 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW024 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW025 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW026 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW027 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW028 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW029 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW030 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW031 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW032 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW033 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ALTUSER AUTOPLEX NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTXCF   NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTXCF2  NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTRPC   NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTBASE  NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
/*
/**********************************************************************
/*  Add the FACILITY class resource profile for the Internal Hardware
/*  Transport
/**********************************************************************
SETROPTS GENERIC(FACILITY)
RDEF FACILITY HSA.ET32*                                  UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
/*
/*********************************************************************
/* Set permissions for Internal Hardware Transport resources
/*********************************************************************
PE HSA.ET32*                CLASS(FACILITY) ID(SYS1)     ACCESS(CONTROL)
/*
@@
//
```

Figure 61. RACF definitions that are required by Internal Hardware Transport

Step 8: Customizing the CNMSTYLE style sheet and members DSIDMNK, DSIOPFU and DSICMPRC

Customizing NetView System Services (NVSS), the backbone of msys for Operations (HPZ8500), involves altering the NVSS Style Sheet and some additional members. The changes to the options are determined by the security scheme chosen and whether the full NetView product is already used by the installation. All members are located in the NETVIEW.DSIPARM data set. The style sheet is *CNMSTYLE* and always requires changing. Optional members are DSIDMNK, DSIOPFU, and DSICMPRC. Copy these members into the shared user-defined DSIPARM data set, locate the relevant record, and make the changes as highlighted. For more information, see "Customizing the initialization style sheet" on page 45.

Customization for NVSS—no NetView license

CNMSTYLE

1. Insert * in the following line:

   ```
   *DOMAIN = C&NV2I.01
   ```

   Add the statement:

   ```
   DOMAIN = &DOMAIN.
   ```

2. Insert * in the following line:

   ```
   *NetID = &CNMNETID.
   ```

3. Insert * in the following line:

   ```
   *SSIname = C&NV2I.CSSIR
   ```

   Add the statement:

   ```
   SSIname = &DOMAIN.SIR
   ```

Note: Changes to the optional members DSIDMNK and DSIOPFU are necessary if an SAF product is *not* being used.

DSIDMNK

1. Remove * from the following line:

   ```
   OPTIONS  OPERSEC=NETVPW,CMDAUTH=SCOPE,OPSPAN=NETV
   ```

2. Insert * in the following line:

   ```
   *OPTIONS  OPERSEC=SAFDEF,CMDAUTH=SAF,AUTHCHK=SOURCEID
   ```

3. Insert * in the following line:

   ```
   *OPTIONS  BACKTBL=CNMSBAK1
   ```

DSIOPFU

Add the following set of lines for each operator, where *opid* is the operator's ID and *password* is their password:

```
opid       OPERATOR    PASSWORD=password
           PROFILEN    DSIPROFA
```

Customization for full NetView 1.4

CNMSTYLE

1. Insert * in the following line:

   ```
   *DOMAIN = C&NV2I.01
   ```

   Add the statement:

   ```
   DOMAIN = &DOMAIN.
   ```

2. Insert * in the following line:
   ```
   *NetID = &CNMNETID.
   ```
3. Insert * in the following line:
   ```
   *SSIname = C&NV2I.CSSIR
   ```
   Add the statement:
   ```
   SSIname = &DOMAIN.SIR
   ```
4. Remove * in front of SA (that is, uncomment SA):
   ```
   TOWER = SA *AON  *MSM  *Graphics *AMI MVScmdMgt
   ```
5. Insert * in the following line:
   ```
   *TOWER.SA  = license
   ```
6. Change Y to N in the following line:
   ```
   TASK.CNMTAMEL.INIT=N
   ```
7. Change Y to N in the following line:
   ```
   TASK.DUIDGHB.INIT=N
   ```
8. Insert * in the following line:
   ```
   *%INCLUDE C&NV2I.STGEN
   ```
9. Change N to Y in the following line:
   ```
   TASK.CNMCALRT.INIT=Y
   ```

DSICMPRC

1. Change Y to N in the following line:
   ```
   EZLSPIPC   CMDMDL   MOD=EZLSPIPC,TYPE=R,RES=N,ECHO=N,SEC=BY
   ```
2. Change Y to N in the following line:
   ```
   EZLSRTVE   CMDMDL   MOD=EZLSRTVE,TYPE=R,RES=N,ECHO=N,SEC=BY
   ```

DSIDMNK

1. Remove * from the following line:
   ```
   LOADEXIT NONE
   ```

   Note: Changes to the optional members DSIDMNK and DSIOPFU are necessary when an SAF product is not being used.
2. Remove * from the following line:
   ```
   OPTIONS  OPERSEC=NETVPW,CMDAUTH=SCOPE,OPSPAN=NETV
   ```
   Insert * in the following line:
   ```
   *OPTIONS  OPERSEC=SAFDEF,CMDAUTH=SAF,AUTHCHK=SOURCEID
   ```
   Insert * in the following line:
   ```
   *OPTIONS  BACKTBL=CNMSBAK1
   ```

DSIOPFU

Add the following set of lines for each operator, where *opid* is the operator's ID and *password* is their password:

```
opid       OPERATOR    PASSWORD=password
           PROFILEN    DSIPROFA
```

Customization for NetView 5.1

When using NetView 5.1 be aware that the DSIDMNK member has been removed. Statements previously in this member are now part of the style sheet. In addition, SAF is not the default security scheme. Because of this there are changes relating to the security setup irrespective of the scheme chosen.

CNMSTYLE

Step 8: Customizing the CNMSTYLE style sheet and members DSIDMNK, DSIOPFU and DSICMPRC

1. Insert * in the following line:

   ```
   *DOMAIN = C&NV2I.01
   ```

   Add the statement:

   ```
   DOMAIN = &DOMAIN.
   ```

2. Insert * in the following line:

   ```
   *NetID = &CNMNETID.
   ```

3. Insert * and add the statement in the following lines:

   ```
   *SSIname = C&NV2I.CSSIR
   *SSIname = C&NV2I.CSSIR
   SSIname = &DOMAIN.SIR
   ```

4. When SAF is being used, insert * in the NETVPW line and remove * from the SAFDEF lines, as follows:

   ```
   *SECOPTS.OPERSEC = NETVPW
   *SECOPTS.OPERSEC = SAFCHECK
   *SECOPTS.OPERSEC = SAFPW
   SECOPTS.OPERSEC = SAFDEF
   *SECOPTS.OPERSEC = MINIMAL
   ```

5. Insert * in TABLE.CNMSCAT2 line and, *only* if SAF is being used, remove * from the SAF.CNMSBAK1 line, as follows

   ```
   *SECOPTS.CMDAUTH = TABLE.CNMSCAT2
   SECOPTS.CMDAUTH = SAF.CNMSBAK1
   *SECOPTS.CMDAUTH = SAF.PASS
   *SECOPTS.CMDAUTH = SAF.FAIL
   *SECOPTS.CMDAUTH = SCOPE.CNMSCOP1
   ```

6. No changes are necessary to the following lines:

   ```
   SECOPTS.AUTHCHK = SOURCEID
   *SECOPTS.AUTHCHK = TARGETID
   ```

7. Insert * in the NETV line and remove * from the SAF line, as follows:

   ```
   *SECOPTS.OPSPAN = NETV
   SECOPTS.OPSPAN = SAF
   ```

8. Remove * from the SA option and add * to the NPDA and NLDM options, as follows:

   ```
   TOWER = SA *AON  *MSM  *Graphics  MVScmdMgt *NPDA  *TARA *NLDM *AMI
   ```

9. Insert * in the following line:

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

10. Change Y to N in the following line:

    ```
    TASK.CNMTAMEL.INIT=N
    ```

11. Change Y to N in the following line:

    ```
    TASK.DUIDGHB.INIT=N
    ```

12. Insert * in the following line:

    ```
    *%INCLUDE C&NV2I.STGEN
    ```

Note: The following changes to options are necessary when an SAF product is not being used.

1. No changes are required to the following lines:

   ```
   SECOPTS.OPERSEC = NETVPW
   *SECOPTS.OPERSEC = SAFCHECK
   *SECOPTS.OPERSEC = SAFPW
   *SECOPTS.OPERSEC = SAFDEF
   *SECOPTS.OPERSEC = MINIMAL
   ```

2. Insert * in TABLE.CNMSCAT2 line and remove * from the SCOPE.CNMSCOP1 line, as follows

```
*SECOPTS.CMDAUTH = TABLE.CNMSCAT2
SECOPTS.CMDAUTH = SAF.CNMSBAK1
*SECOPTS.CMDAUTH = SAF.PASS
*SECOPTS.CMDAUTH = SAF.FAIL
SECOPTS.CMDAUTH = SCOPE.CNMSCOP1
```

3. These are the defaults, so no changes are necessary to the following lines:

```
SECOPTS.AUTHCHK = SOURCEID
*SECOPTS.AUTHCHK = TARGETID
```

4. These are the defaults, so no changes are necessary to the following lines:

```
SECOPTS.OPSPAN = NETV
*SECOPTS.OPSPAN = SAF
```

DSIOPFU

Add the following set of lines for each operator, where *opid* is the operator's ID and *password* is their password:

```
opid      OPERATOR   PASSWORD=password
          PROFILEN   DSIPROFA
```

Step 9: Customizing the member AOFCUST

Turn on the msys for Operations functions that you are interested in. By default all functions that take automated action ship ***disabled*** and are controlled by the AOFCUST member located in ING.SINGNPRM. Extensive comments are included within AOFCUST. Start by copying this member to the shared DSIPARM data set previously created and make the desired changes there. For more information, see "Editing the customization member AOFCUST" on page 46.

The ***AUTO*** section controls function enablement. Functions are enabled by removing comments (*) from the statements you wish to enable. Each function has a corresponding *detailed section* where installation specific values are coded. Such as the actions you permit msys for Operations to take, volumes available for dynamic allocation, data set HLQs and so on.

Processor and logical partition configurations are also defined to msys for Operations within this member. This is done in the HW Section which has no relationship to the AUTO section. During msys for Operations initialization checks are made to see if z/OS is able to use an Internal Hardware Transport for direct hardware interaction. If so, this capability is enabled.

Note: It is critical that everything defined in the HW section exactly represents the definitions in HCD and in use by the Support Element (SE) for any given processor. Failure to do this could result in hardware manipulation that targets the wrong CPC/LPAR.

```
*---------------------------------------------------------------
* AUTO section
*---------------------------------------------------------------
* This section contains keywords representing automated
* functions:
*   CDS       Enables recovery of missing CDS allocations.
*   HEALTHCHK Enables viewing of Best Practices values as compared
*             to User defined values. This assists in reducing multi-
*             system outages (MSO).
*   ENQ       Enables recovery of a long running ENQ detection and
*             "HUNG" command recovery.
*   LOG       Enables recovery of a syslog start failure.
*   LOGGER    Enables recovery of a system logger offload
*             condition.
*   PAGE      Enables recovery of a local page dataser shortage.
*   WTO       Enables recovery of WTO/WTOR buffer shortage
*             conditions.
*   XCF       Enables recovery to prevent a sysplex outage when a
*             system leaves the sysplex due to a failure condition.
* Other keywords are not valid.
* These keywords only affect message initiated automation.
* They do not affect automation initiated using INGCF/INGPLEX.
*
* An asterisk '*' placed in column 1 marks a line as comment.
* Example:To enable the WTO/WTOR Buffer Shortage Recovery use:
*
* AUTO(
*  WTO
* )
*
* or
*
* AUTO(
* * CDS
* * HEALTHCHK
* * ENQ
* * LOGGER
* * LOG
* * PAGE
*   WTO
* * XCF
* )
*---------------------------------------------------------------
*
AUTO(
* CDS
* HEALTHCHK
* ENQ
* LOG
* LOGGER
* PAGE
* WTO
* XCF
)
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 1 of 17)

```
*---------------------------------------------------------------
* COMMON section
*---------------------------------------------------------------
* The definitions within this section are common to all other
* sections.
* Do NOT comment out this section.
*
* The parameters in this section and all following sections may use
* MVS or user-defined system symbolics, including the NetView-supplied
* &DOMAIN symbolic.
*
* The following parameters must be defined per line.
*
* 1.The keyword TEMPHLQ.
*
*   TEMPHLQ: This keyword introduces a high level qualifier
*            which is used to assemble a data set name for
*            allocating temporary data sets needed by programs
*            running as started tasks.
*            The qualifier may consist of up to 17 characters
*            according to the OS/390 data set naming rules
*            (hlq1.hlq2.hlq3).
*            There must be only one line containing the TEMPHLQ
*            keyword within AOFCUST.
*
*            Note: Netview must have RACF ALTER access to the
*                  qualifier. The userid of the started tasks
*                  must have RACF UPDATE access to the qualifier.
*
* 2.The keyword STCJOBNM
*
*   STCJOBNM: This keyword introduces the job name being used
*             for programs running as started tasks.
*             The qualifier may consist of up to 8 characters
*             according to the OS/390 job naming rules.
*             There must be only one line containing the
*             STCJOBNM keyword within AOFCUST.
*             When not defined the job name of each started task
*             defaults to the procedure name.
*
* Example:
*
* COMMON(
*   TEMPHLQ  (MSYS.TEMP.DSNHLQ)
*   STCJOBNM (MSYSSTCJ)
* )
*--------------------------------
*
COMMON(
  TEMPHLQ (MSYS.TEMP)
)
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SIGNPRM (Part 2 of 17)

```
*---------------------------------------------------------------
* WTOBUF section
*---------------------------------------------------------------
* The definitions within this section will be applied if 'WTOBUF'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
* With each statement you can define address spaces handled
* separately at the resolution of WTO/WTOR buffer shortage
* conditions.
* Three parameters must be defined per line.
* 1.The address space name. A wildcard character (i.e.'*') is
*   supported at the end of this parameter.
* 2.WTO,WTOR or *. This parameter indicates if the automation
*   shall be applied to WTO buffer shortage conditions,
*   or to WTOR buffer shortage conditions
*   or to both.
* 3.KEEP or CANCEL. This parameter indicates if an address
*   space must be kept or canceled upon a buffer shortage
*   condition which is caused by this address space.
*
* Note:1. The default for all job names not listed in this section
*      is KEEP.
*      2. If there are multiple statements within the
*      WTOBUF section they either should define all
*      job names of which address spaces to be kept (KEEP) or
*      they should define only those to be canceled (CANCEL).
*      3. '* * CANCEL' will override the default to cancel all
*      address spaces. It should be the only statement
*      within the WTOBUF section.
*
* If AOFCUST is shared between the images within a sysplex,
* these definitions are valid sysplex wide.It must be considered
* that the jobnames for your applications can be different on
* each system.
*
*
* Example:Applications with jobname CICS and jobname beginning
*         with IMS must not be canceled (must be kept) upon
*         WTO and WTOR buffer shortage conditions.
*         All other jobs can be canceled.
*
* WTOBUF(
*   CICS   *   KEEP
*   IMS*   *   KEEP
* )
*---------------------------------------------------------------
*
WTOBUF(
*JOB1     WTOR CANCEL
*JOB2*    WTO  CANCEL
*JOB3*    *    CANCEL
*         WTOR CANCEL
*         WTO  CANCEL
)
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 3 of 17)

```
*--------------------------------------------------------------
* CDS section
*--------------------------------------------------------------
* The definitions within this section will be applied if 'CDS'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section CDS related automation can be customized.
* Two parameters must be defined per line.
*
* 1.The keyword HLQ or VOL.
*
*   HLQ:This keyword introduces a high level qualifier
*       which is used to assemble a data set name for
*       creating and/or allocating an alternate CDS.
*       The qualifier may consist of up to 26 characters
*       according to the OS/390 data set naming rules
*       (hlq1.hlq2.hlq3).
*       The qualifier is appended by the CDS type and '.CDS0n'
*       where 'n'is a sequence number.
*       There must be only one line containing the HLQ keyword
*       within AOFCUST. The HLQ keyword must be the first word
*       in line.
*
*   VOL:This parameter introduces a list of volume names
*       per CDS type. The list contains the names of volumes
*       which are eligible when automation is going to
*       creating and/or allocate an alternate CDS.
*       There may be multiple 'VOL'definitions but only one
*       per type and per line.
*       The VOL keyword must be the first word in line.
*
* 2.The list of volume names per CDS type
*
*   (cds_type,vol1,vol2,vol3,vol4,vol5,vol6,vol7,vol8)
*
*     'cds_type' represents the CDS type.
*     In case of automation an alternate CDS of this type
*     will be allocated.
*     Valid values are SYSPLEX,ARM,CFRM,LOGR or SFM.
*
*     'vol1,vol2,vol3,...' represent volume names.
*     The volume names must be in accordance to the OS/390
*     volume naming rules.
*     A maximum of 8 volumes can be defined.
*
* Note: Do not specify SMS managed volumes.
*       Do not specify volumes which already contain allocated
*        couple data sets.
*
*
* Example:
*
* CDS(
*   HLQ SYS1.PLEX2.CDS
*   VOL (SYSPLEX,CDS01,CDS02,CDS03)
*   VOL (CFRM,CDS01,CDS02,CDS03)
*   VOL (ARM,CDS01,CDS02,CDS03)
*   VOL (LOGR,CDS01,CDS02,CDS03)
*   VOL (SFM,CDS01,CDS02,CDS03)
* )
*--------------------------------
*
CDS(
  HLQ  hlq1.hlq2.hlq3,
* VOL  (CFRM,vol1),
* VOL  (ARM,vol2,vol3),
* VOL  (LOGR,vol4,vol5,vol6),
* VOL  (SFM,vol7,vol8,vol9,vola),
* VOL  (SYSPLEX,volb,volc,vold,vole,volf,volg,volh,voli)
)
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 4 of 17)

```
*----------------------------------------------------------------
* ENQ section
*----------------------------------------------------------------
* The definitions within this section will be applied if 'ENQ'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* Note that a customization of the "HUNG" command recovery is not
* possible.
*
* In this section ENQ related automation can be customized.
*
* 1.The keywords DUMP, JOB, RES, SYMDEF and TITLE.
*
*   DUMP:  This keyword defines the DUMP option being used
*          for the SDATA parameter on the dump command. It
*          applies for the JOB(job,DUMP) statements only.
*          (Range: Any combination of SDATA dump values)
*
*   JOB:   This parameter defines what jobs cannot be cancelled
*          (JOB(job,KEEP)), what jobs can be cancelled without a
*          dump (JOB(job,NODUMP)), what jobs can be cancelled
*          with a dump using the default dump options
*          (JOB(job,DUMP), and finally what jobs can be
*          cancelled with a dump using the IEADMCxx PARMLIB
*          members.
*          (Range: jobnames including wild cards or 4 character
*                  address space ids)
*
*   RES:   This parameter defines the major and minor resources
*          being checked for long running ENQs. The time value
*          is the time after which an ENQ is treated as long running
*          Or, if you specify any abbreviation of EXCLUDE this
*          resource is not monitored at all. However, any exclude
*          definition requires a generic include definition
*          RES(*,*,time) to complete the exclusion list.
*          (Range: 1 to 8  characters major resource name,
*                  1 to 50 characters minor resource name,
*                  Exclude or 30 to 999 seconds wait time)
*
*   SYMDEF:This parameter defines a system sysmbol and its
*          substitution value.
*          (Range: Any valid symbol definition)
*
*   TITLE: The parameter defines the title of each dump being
*          taken with the default dump options.
*          (Range: Up to 100 characters in mixed case)
*
*   JOB, RES, and SYMDEF statements can be defined as much as
*   needed. For DUMP, SYMDEF, and TITLE statements refer to
*   the MVS DUMP command for more details.
*
* 2.The default values for parameters being omitted
*
*     DUMP:  CSA GRSQ RGN SQA NOSUM TRT
*
*     JOB:   *,DUMP
*            Only, when no "JOB(*,..)" statement has been defined
*
*     TITLE: Dump by msys for Operations due to a lonq ENQ detection
*
*     RES:   RES(*,*,30)
*            is automatically added if you specify any
*            RES(major,minor,Exclude) statement but no
*            RES(*,*,time) statement.
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 5 of 17)

```
* Example:
*
* ENQ(
*   DUMP (sdata_values)
*   JOB (0001,KEEP)
*   JOB (CICS*,KEEP)
*   JOB (IMS*,NODUMP)
*   JOB (ABC*,D0,D1)
*   JOB (*,DUMP)
*   RES (MAJOR1,MINOR1,EXCL)
*   RES (MAJOR1,MINOR*,60)
*   RES (MAJOR2,*,120)
*   RES (MAJOR3*,*,300)
*   RES (*,*,999)
*   SYMDEF (*,&DUMPID01='GLOBAL')
*   SYMDEF (*,&DUMPID02='LOCAL')
*   TITLE (Dump by automation due to a long ENQ detection)
* )
*--------------------------------
*
ENQ(
*  RES    (*,*,999)
*  DUMP   (CSA,GRSQ,RGN,SQA,NOSUM,TRT)
*  JOB    (*,DUMP)
*  TITLE  (Dump by msys for Operations due to a lonq ENQ detection)
)
*
*----------------------------------------------------------------
* PAGE section
*----------------------------------------------------------------
* The definitions within this section will be applied if 'PAGE'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section PAGE related automation can be customized.
*
* 1.The keywords CYL, DSN, HLQ, JOB, and VOL.
*
*   CYL:This parameter defines the maximum number of cylinders
*       used for the space allocation to format a local page
*       data set dynamically. The minimum/maximum/default values
*       are 100/999/400. The recovery uses the maximum
*       available space between the minimum and the specified
*       value.
*       Note: On a 3390 DASD 100 CYLS are adequate to 70 MB.
*             The formatting process lasts approximately 18
*             seconds for this amount of space.
*
*   DSN:This parameter defines a pre-formatted spare page
*       data set to be used in the recovery situation.
*       Note: The data set must be allocated on a volume
*             shared by all systems in the sysplex.
*
*   HLQ:This keyword introduces a high level qualifier
*       which is used to assemble a data set name for
*       creating and allocating a page data set.
*       The qualifier may consist of up to 23(!) characters
*       according to the OS/390 data set naming rules
*       (hlq1.hlq2.hlq3).
*       The qualifier is appended the system name followed by
*       '.Vvvvvvv.Snn' where 'v' is the volume serial number and
*       'n' sequence number from 00 to 99.
*       Note: The high level qualifier must point to the master
*             catalog and must not be SMS managed.
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 6 of 17)

```
*   JOB:This parameter defines what jobs cannot be cancelled
*       (KEEP) or can be cancelled (CANCEL) in case the
*       auxiliary shortage condition cannot be resolved and
*       the job is one of those jobs with the most rapidly
*       increasing storage requirements.
*       Trailing wild card is supported.
*
*   VOL:This parameter introduces a list of volume names which are
*       eligible when automation creates and allocates a new page
*       data set. The volume names must be in accordance to the OS/390
*       volume naming rules. This parameter must be used in conjunction
*       with the HLQ parameter.
*       Note: The volume must be shared by all systems in the sysplex.
*
*   All keywords except HLQ can be defined as many as needed.
*   At least one of DSN or HLQ/VOL must be defined.
*
* 2.The default values for parameters being omitted
*
*   CYL:   400
*
*   JOB:   *,KEEP
*          Only, when no "JOB(*,..)" statement has been defined
*
* Example:
*
* PAGE(
*   DSN (dsn1)
*   DSN (dsn2)
*   HLQ (hlq1.hlq2.hlq3)
*   CYL (nnn)
*   JOB (ABC,KEEP)
*   JOB (ABC*,CANCEL)
*   JOB (*,KEEP)
*   VOL (vol111,vol112,vol113)
*   VOL (vol211)
*   VOL (vol311,vol312)
* )
*--------------------------------
*
PAGE(
* HLQ (hlq1.hlq2.hlq3)
* CYL (400)
* VOL (vol1,vol2)
* JOB (*,KEEP)
* DSN (dsn1)
* DSN (dsn2)
)
*
*---------------------------------------------------------------
* HW section
*---------------------------------------------------------------
* The definitions within this section will be applied indepent
* of the definitions in the AUTO section. They apply to coupling
* facility related functions as well as to the automation of the
* message IXC102A. The defintions must reflect the actual
* hardware configuration. Otherwise the appropriate functions
* will not work properly.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section HW related automation can be customized.
*
* 1.The keywords CPC and IMAGE.
*
*   CPC:  This parameter defines a CPC (Central Processor
*         Complex).
*
*   IMAGE:This parameter defines an operating system running on
*         a CPC. To indicate that CPC runs in basic mode you
*         must define the CPC name as the LPAR name.
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 7 of 17)

```
*   You can define as many CPC and IMAGE statements as needed
*   but only one per line.
*
* 2.The parameters in particular:
*
*   authcomm  - the authentication value being used for
*               communication with the Support Element.
*               Note: This value must match the value specified
*                     in the Support Element for SNMP communication.
*                     Keep in mind that SNMP handles upper case
*                     and lower characters differently.
*
*   cpcname   - the name of CPC hosting the operating system.
*
*   imagetype - the type of the operating system, either CF,
*               MVS, or OTHER.
*
*   lparname  - the name of LPAR running the opearing system.
*               Note: When a CPC runs in basic mode specify the CPC
*                     name as the LPAR name.
*
*   netid.nau - the network address of the Support Element of
*               the CPC.
*
*   plexname  - the name of the sysplex.
*
*   sysname   - the name of the operating system.
*
*
* Example:
*
* HW(
*   CPC   (cpcname,netid.nau,authcomm)
*   IMAGE (sysname,lparname,cpcname,plexname,imagetype)
* )
*--------------------------------
*
HW(
* CPC   (cpcsyn,netid.nau,authcomm)
* IMAGE (cfname ,lparname,cpcsyn,sysplexname,CF)
* IMAGE (sysname,lparname,cpcsyn,sysplexname,MVS)
* IMAGE (name   ,lparname,cpcsyn,           ,OTHER)
)
*
*---------------------------------------------------------------
* IXC102A section
*---------------------------------------------------------------
* The definitions within this section will be applied if 'XCF'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section IXC102A related automation can be customized.
*
* 1.The keywords CMD, DISABLE, and ENABLE.
*
*   CMD:    This parameter defines the command being sent to the
*           Support Element when the IXC102A message is trapped.
*           Even the Support Element accepts many commands only
*           the next four commands are accpeted by the automation
*           to solve the recovery situation:
*
*             ACTIVATE [PN(image_profile_name)]
*
*             DEACTIVATE
*
*             SYSRESET [CLEAR]
*
*             LOAD [P(load_profile_name)] [CLEAR]
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 8 of 17)

```
*           As the Support Element treats the affected system
*           still as running the automation automatically appends
*           each command with the parameter FORCE forcing the
*           Support Element to accept the command anyway.
*
*   DISABLE:This parameter defines the system(s) being excluded
*           from automation when an IXC102A message is issued for
*           the system.
*
*   ENABLE: This parameter defines the system(s) being automated
*           when an IXC102A message is issued for the system.
*
*
*   You can define DISABLE and ENABLE statements as many as needed
*   but only one per line.
*   You can also have multiple 'CMD' definitions but only one per
*   system and per line.
*
*
* 2.The parameters in particular:
*
*   image_profile_name - the IMAGE profile name designates
*                        information stored in the Support Element
*                        used to build a logical partition in a
*                        CPC and to IPL an operating system
*
*   load_profile_name  - the LOAD profile name designates
*                        information stored in the Support Element
*                        used to IPL an operating system
*
*   CLEAR              - clears the main storage
*
*
*   When you omit the profile name the Support Element uses the
*   profile which was used at the last operation.
*
*
* 3.The default values for parameters being omitted
*
*   The default command for a system defined in the HW section
*   is 'SYSRESET CLEAR'.
*
*
*
* Example:
*
* IXC102A(
*   CMD     (sysname5,'command')
*   CMD     (sysname6,'command')
*   DISABLE (sysname1,sysname2,sysname3)
*   DISABLE (sysname4)
*   ENABLE  (sysname5,sysname6)
*   ENABLE  (sysname7)
* )
*---------------------------------
*
IXC102A(
* CMD     (sysname,SYSRESET CLEAR)
* DISABLE (sysname)
* ENABLE  (sysname)
)
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 9 of 17)

```
* ------------------------------------------------------------------
* Health Checker section
* ------------------------------------------------------------------
* This is the definition of the user's best practices (USERPARM) for
* the Health Checker.
*
* In the following, IBMPARM means the IBM best practices as shown
* by command INGPLEX BESTpractices.
*
* USER INSTRUCTIONS:
* 1. Use USERPARM to override IBMPARM.
* 2. Follow documentation below for details of each check.
* 3. Specify PARMS, TIMEINT, SEVERITY, or NOCALL,
*    as well as REASON, and DATE.
*    DATE in USERPARM should be later than DATE in IBMPARM,
*    REASON in USERPARM should be different from REASON in IBMPARM.
*
* SYNTAX:
* 1. 1st keyword must be CHECK
* 2. End each statement with a semi-colon.
* 3. Use cols 1-72 only for statements.
* 4. Keywords on each statements are:
*    CHECK    - must match a CHECK in IBMPARM.
*    REASON   - max 100 chars - enclose in quotes. For formatting
*               reasons it is recommended that individual words not
*               span lines because a blank is inserted automatically
*               with each line break.
*               Internal processing reduces a sequence of blanks to one
*               blank, so for each such sequence reduce the allowed
*               maximum of 100 characters by 1.
*               If REASON spans multiple lines, then for each but the
*               last line subtract one inserted blank per line from the
*               allowed maximum of 100 characters.
*    DATE     - yyyymmdd - must be later than matching date in IBMPARM.
*               Since the value specified is used for a simple compare
*               with IBM's date, the input values are checked for
*               validity only as far as needed, i.e. not for total
*               match with the calendar, i.e. 20030231 would be
*               accepted.
*    SEVERITY - HIGH, MEDIUM, or LOW - overrides the predefined
*               severity used to flag report data for the check
*    TIMEINT  - hh:mm - overrides the predefined time interval in
*               which the check is repeated - in hours and minutes.
*    NOCALL   - eliminates a check.
*               If you specify both, NOCALL and PARMS, the check is
*               still eliminated
*    PARMS    - format varies with each check - see IBMPARM.
**------------------------------------------------------------------
*
* The following are the definitions from IBMPARM, they can be used as
* samples to set USERPARM.
*
*HEALTHCHK(
*
* ------------------------------------------------------------------
*      CouplingFacility_Structure status:
* ------------------------------------------------------------------
*
*   Create a list of all Systems defined and report on their status.
*
*   Create a list of all CFs defined and report on their status.
*
*   Create a list of all STRuctures defined and report on status.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(CouplingFacility_Structure)
*       Severity(Medium)
*       DATE(20030102)
*       reason('Check CF and Structure location');
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 10 of 17)

```
* -------------------------------------------------------------------
*      CouplingFacility_Structure descriptions:
* -------------------------------------------------------------------
*
*   Create a report showing what systems, CFs, and structures are
*   defined in the sysplex. This report will show appropriate status
*   and connection status of these resources.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(Sys_CF_STR_Report)
*       Severity(Low)
*       DATE(20030102)
*       reason('Create System, CF, Structure report');
*
* -------------------------------------------------------------------
*   XCF_Signalling checks:
* -------------------------------------------------------------------
*   Check #1:
*      Check that ALL transport classes are set up to service
*      the pseudo-group name 'UNDESIG ',
*      i.e. that any XCF message can use each transport class.
*   Check #2:
*      Check that all defined Transport Classes are assigned to
*      at least one pathout (outbound path).
*   Check #3:
*      Check that most pathouts have a transport class defined
*      with a "small" (parm4) classlength and at least one other
*      transport class is defined with a higher "large" (parm5)
*      classlength.
*   Check #4:
*      Check that multiple (at least parm3) Pathout/Pathin pairs
*      are in the WORKING (i.e. OPERATIONAL) state for each system
*      in the sysplex connected to the current system.
*   Check #5:
*      Check that there is a MAXMSG of at least the indicated
*      minimum value (parm1) for each transport class.
*   Check #6:
*      Check each inbound signal path and ensure that each can
*      support at least the indicated minimum number (parm2) of
*      messages from the sending system.
*      (AMDPMXMS / (AMDPATH1_BuffLen + 2K)) should be > specified
*      minimum number of messages supported by the path.
*
* PARAMETERS:
*   Check #1: None required.
*   Check #2: None required.
*   Check #3: Parameters 4 and 5
*   Check #4: Parameter 3
*   Check #5: Parameter 1
*   Check #6: Parameter 2
*
*   XCF_Signalling positional parameter descriptions:
*
*   1. Check #5 parameter 1:
*      The minimum MAXMSG value for transport classes.
*      This is an INTEGER. The maximum acceptable value is 999999.
*
*   2. Check #6 parameter 2:
*      The minimum number of XCF messages that an inbound
*      XCF signal path should support to avoid message backup.
*      This is an INTEGER. The maximum acceptable value is 999999.
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 11 of 17)

```
*   3. Check #4 parameter 3:
*      Specifies the minimum pathout/pathin pair count
*      for a system.
*      This is an INTEGER. The maximum acceptable value is 9.
*
*   4. Check #3 parameter 4:
*      Specifies the maximum value to be interpreted as a "small"
*      (XCF transport) classlength.
*      This is an INTEGER. The maximum acceptable value is 9999.
*
*   5. Check #3 parameter 5:
*      Specifies the minimum value to be interpreted as a "large"
*      (XCF transport) classlength.
*      This is an INTEGER. The maximum acceptable value is 62464.
*                          The minimum acceptable value is 4028.
*      The specified value does not include the 68 additional bytes
*      used by XCF for internal control blocks
* ------------------------------------------------------------------
*CHECK(XCF_Signalling)
*       Severity(Medium)
*       DATE(20030102)
*       PARMS(750,30,2,956,4028)
*       REASON('Avoid problems with XCF signalling.');
*
* ------------------------------------------------------------------
*   XCF signalling structures in coupling facilities:
* ------------------------------------------------------------------
*   When XCF signalling structures in coupling facilities are used,
*   check that:
*     1. not all the signalling structures reside on the same
*        coupling facility (CF).
*     2. multiple links (or CHPIDs) to each CF are both
*        ONLINE and OPERATING.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(XCF_Signalling_Structures_in_CF)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Avoid problems with XCF signalling in CFs.');
*
* ------------------------------------------------------------------
*    CONSOLE Names:
* ------------------------------------------------------------------
* Check that each Console has the NAME parameter specified
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Console_Names)
*       Severity(High)
*       DATE(20030102)
*       REASON('Like named consoles are matched across the sysplex');
*
* ------------------------------------------------------------------
*  Alternate CONSOLE Groups:
* ------------------------------------------------------------------
* Check that each Console has the ALTGRP parameter specified.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Alternate_Console_groups)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Provides good recovery from console loss');
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 12 of 17)

```
* -------------------------------------------------------------------
*  Master authority:
* -------------------------------------------------------------------
* Check that each system has a console with MASTER authority.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(Console_master)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Needed for DCCF and other situations');
*
* -------------------------------------------------------------------
*   CONSOLE MSCOPE versus Routcodes:
* -------------------------------------------------------------------
* Check that each console has an acceptable mix of MSCOPE and
* Routcodes.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(Console_MSCOPE_and_Routcodes)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Avoids overloading any console. Reduces number of
*messages sent to Sysplex consoles');
*
* -------------------------------------------------------------------
*  AMRF and Eventual Action message retention:
* -------------------------------------------------------------------
* If AMRF is ON, check that Eventual_Action messages are not
*    retained.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(AMRF_and_MPF_consistent)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('Avoids long chains of messages in storage');
*
* -------------------------------------------------------------------
*  CONSOLEs and ROUTCODE 11:
* -------------------------------------------------------------------
* Check that no console is receiving ROUTCODE 11 messages.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(Console_routcode_11)
*       Severity(Low)
*       DATE(20030102)
*       REASON('Not really needed as for programmer info only');
*
* -------------------------------------------------------------------
*  Sysplex Master Console:
* -------------------------------------------------------------------
* Check that the MASTER console is active.
*
* PARAMETERS: None required.
* -------------------------------------------------------------------
*CHECK(Sysplex_master)
*       Severity(High)
*       DATE(20030102)
*       REASON('Needed in emergencies');
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 13 of 17)

```
*  EMCS consoles and MSCOPE versus Routcodes:
* ------------------------------------------------------------------
* Check that each EMCS console has an acceptable mix of MSCOPE and
* Routcodes.
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(EMCS_MSCOPE_and_Routcodes)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('ROUTCODE(ALL) and non-local MSCOPE will cause a large
*number of messages to be processed');
*
* ------------------------------------------------------------------
*  EMCS Consoles and HARDCOPY Flag:
* ------------------------------------------------------------------
* Check that each EMCS console does not have the HARDCOPY flag set
* if MSCOPE > 1.
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(EMCS_hardcopy)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('EMCS consoles with HARDCOPY specified will process an
*excessive number of messages');
*
* ------------------------------------------------------------------
*  SYSCONS and MSCOPE:
* ------------------------------------------------------------------
* Check that the SYSCONS has only local MSCOPE.
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(SYSCONS_MSCOPE)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('If SYSCONS is used in emergencies it should not have
*to process large numbers of messages');
*
* ------------------------------------------------------------------
*  SYSCONS and ROUTCODE:
* ------------------------------------------------------------------
* Check that the SYSCONS has advisable routcodes.
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(SYSCONS_ROUTCODES)
*       Severity(Low)
*       DATE(20030102)
*       REASON('If SYSCONS is used in emergencies it should not have
*to process large numbers of messages');
*
* ------------------------------------------------------------------
*  Number of EMCS Consoles
* ------------------------------------------------------------------
* Check that there is not an excessive number of EMCS consoles.
*
* PARAMETERS:
*       1. Maximum number of ACTIVE EMCS consoles on this system.
*          Values between 0 and 99999999 are accepted.
*          Must be numeric.
*       2. Maximum number of INACTIVE EMCS consoles on the entire
*          sysplex. Values between 0 and 99999999 are accepted.
*          Must be numeric.
* ------------------------------------------------------------------
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 14 of 17)

```
*CHECK(Number_EMCS_consoles)
*       Severity(High)
*       DATE(20030102)
*       PARMS(5000,10000)
*       REASON('Excessive numbers of EMCS consoles cause slowdown');
*
* ------------------------------------------------------------------
*  SYS CONS and PD mode:
* ------------------------------------------------------------------
* Check that SYSCONS is not in PD mode
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(SYSCONS_PD_MODE)
*       Severity(Low)
*       DATE(20030102)
*       REASON('SYSCONS should be run in Problem Determination mode
*only when there is a problem');
*
* ------------------------------------------------------------------
*  SYSCONS and MASTER authority:
* ------------------------------------------------------------------
* Check that SYSCONS has MASTER authority
*
* PARAMETERS: None
* ------------------------------------------------------------------
*CHECK(SYSCONS_MASTER)
*       Severity(High)
*       DATE(20030102)
*       REASON('SYSCONS needs MASTER authority to resolve problems in
*emergency situations');
*
* ------------------------------------------------------------------
*  Available Frame Queue Thresholds:
* ------------------------------------------------------------------
* Check that the available frame queue thresholds are not set too
* low.
*
* PARAMETERS:
*       1. 64 bit Minimum LOW threshold (special action commences).
*       2. 64 bit Minimum OK threshold (special action ceases).
*       3. 31 bit Minimum LOW threshold (special action commences).
*       4. 31 bit Minimum OK threshold (special action ceases).
* ------------------------------------------------------------------
*CHECK(Available_Frame_Queue_Thresholds)
*       Severity(High)
*       DATE(20030211)
*       PARMS(400,600,200,400)
*       REASON('System may not recover in time if set too low');
*
* ------------------------------------------------------------------
*  V=R specification:
* ------------------------------------------------------------------
* Check for the existence of V=R (REAL) storage.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(Real_Storage_Availability)
*       Severity(Low)
*       DATE(20030102)
*       REASON('Performance may be impacted');
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 15 of 17)

```
* ------------------------------------------------------------------
*  Reconfigurable Storage specification:
* ------------------------------------------------------------------
* Check for the existence of reconfigurable (RSU) storage.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(RSU_Storage_Availability)
*       Severity(Low)
*       DATE(20030102)
*       REASON('Performance may be impacted');
*
* ------------------------------------------------------------------
*   XCF Cleanup value:
* ------------------------------------------------------------------
* Check that the XCF cleanup time is set to a reasonable value to
* hasten the removal of a dead system from the SYSPLEX.
*
* PARAMETERS:
*       1. Recommended XCF cleanup time in seconds.
*          The maximum acceptable value is 86400
* ------------------------------------------------------------------
*CHECK(XCF_Cleanup_Value)
*       Severity(Low)
*       DATE(20030102)
*       PARMS(15)
*       REASON('Quick removal of a dead system from SYSPLEX');
*
* ------------------------------------------------------------------
*  XCF Failure Dectection Interval setting:
* ------------------------------------------------------------------
* Check that the XCF failure detection interval equates to the
* formula PARM1*SPINTIME+PARM2.
*
* PARAMETERS:
*       1. Multiplier.
*       2. Constant.
* ------------------------------------------------------------------
*CHECK(XCF_Failure_Detection_Interval)
*       Severity(Medium)
*       DATE(20030102)
*       PARMS(2,5)
*       REASON('Allow adequate time to recover from spin situation
*before system is assumed dead');
*
* ------------------------------------------------------------------
*  Sysplex Failure Management:
* ------------------------------------------------------------------
* Check that the status of a SYSPLEX failure management (SFM) policy
* is as recommended.
*
* PARAMETERS:
*       1. Recommended SFM status (ACTIVE/INACTIVE).
* ------------------------------------------------------------------
*CHECK(XCF_SYSPLEX_Failure_Management)
*       Severity(Medium)
*       DATE(20030102)
*       PARMS(ACTIVE)
*       REASON('An SFM policy provides better failure management');
*
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 16 of 17)

```
* ------------------------------------------------------------------
*  SDUMP dynamic allocation of datasets:
* ------------------------------------------------------------------
* Check that SDUMP is using dynamic allocation and that it has
*  not been disabled by the CHNGDUMP command.
*
* PARAMETERS: None required.
* ------------------------------------------------------------------
*CHECK(SDUMP_Availability)
*       Severity(Medium)
*       DATE(20030102)
*       REASON('SDUMP setup should ensure adequate diagnostics are
*gathered on the 1st occurrence of problems');
*
* ------------------------------------------------------------------
*  GRS mode:
* ------------------------------------------------------------------
* Check that GRS is in the suggested mode
*
* PARAMETERS:
*       1. Mode required, STAR, RING or NONE.
* ------------------------------------------------------------------
*CHECK(GRS_Mode)
*       Severity(High)
*       DATE(20030102)
*       PARMS(STAR)
*       REASON('GRS should run in STAR mode to improve performance.')
*
* ------------------------------------------------------------------
*  Couple Dataset Separation:
* ------------------------------------------------------------------
* Check that SYSPLEX Couple dataset and Function Couple datasets
*  are properly isolated with alternates.
*
* Parameters:  N/A
* ------------------------------------------------------------------
*CHECK(CDS_Dataset_Separation)
*       Severity(High)
*       DATE(20030102)
*       reason('Ensure that CDS separation has been maintained');
*
* ------------------------------------------------------------------
*  Filesystem Automove setting:
* ------------------------------------------------------------------
* Check that Unix System Services Filesystem Automove is correct
*       in a SYSPLEX environment
* PARAMETERS:
*       1. File system MODE, SYSPLEX has Automove support
*          other File Modes, NOPLEX will not check AutoMove
* ------------------------------------------------------------------
*CHECK(USS_FILESYS_CONFIG)
*       Severity(High)
*       DATE(20030102)
*       PARMS(SYSPLEX)
*       REASON('USS Automove moves a file system to a new system in
*the Sysplex when the owning system fails');
*)
HEALTHCHK(
*CHECK(GRS_Mode)
*       DATE(my date)
*       PARMS(STAR)
*       TIMEINT(24:00)
*       SEVERITY(High)
*       REASON('my reason');
)
```

Figure 62. AUTO section of the AOFCUST member in ING.SINGNPRM (Part 17 of 17)

Step 10: Building the VTAM logmode table

This step is optional depending on whether prior installation actions have been taken. If the installation is currently running NetView as part of the normal process, then the actions described here will most likely have been done. If not, perform these additional tasks to build the VTAM logon mode table, AMODETAB. This table defines the session protocols for the different devices and applications used by msys for Operations. For more information, see “Changing the logmode table (MODETAB parameter)” on page 44.

1. Start off by checking whether AMODETAB is already in place. Browse the JCL statements used to start VTAM. Locate the VTAMLIB DD definition statement which may address a single data set or multiple data sets that are concatenated together. Browse the applicable data sets checking for a member AMODETAB. If found browse the actual member and issue a FIND DSIL6MOD browse command. If everything checks out, you are done. If nothing is found, proceed with the creation of this member.

   Note: If AMODETAB is found but an entry for DSIL6MOD is not, then steps will need to be taken in conjunction with the installation’s Networking Group to modify the current AMODETAB to include the DSIL6MOD statements referenced below.
2. Prepare a data set into which AMODETAB will be linked. Although this can be done directly into SYS1.VTAMLIB, the recommendation is to create a user defined VTAMLIB data set, with the same attributes as SYS1.VTAMLIB and make this the first data set in the VTAMLIB concatenated list. If a user defined VTAMLIB data set already exists, then you are done here and can proceed to the next step.

   Note: Remember that SYS1.VTAMLIB is authorized. Any new data set that is made part of the same concatenation must also be authorized. Refer to Step 2: Adding additional procedures to PROCLIBon how to do this.
3. Compile and linked it AMODETAB into the chosen data set. Two sample members, CNMSJ006 (JCL) and CNMS0001 (AMODETAB source), can be found in NETVIEW.CNMSAMP. The following job, is based on CNMSJ006 but reduced to the specific statements required. Make similar changes applicable to your installation and run the job to create the AMODETAB member. If the SYSLMOD data set that you chose already existed and was part of the procedure used to start VTAM, you are done. Otherwise, complete the final task.

   ```
   //AMODETAB JOB (034D000,TS),NORTHRUP,CLASS=C,MSGCLASS=T,
   //             REGION=6M,NOTIFY=&SYSUID
   //ASM     EXEC PGM=ASMA90,PARM='NODECK,OBJECT'
   //SYSPRINT DD  SYSOUT=*
   //SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
   //         DD  DSN=SYS1.SISTMAC1,DISP=SHR
   //SYSUT1   DD  UNIT=3390,SPACE=(CYL,(1,1))
   //SYSUT2   DD  UNIT=3390,SPACE=(CYL,(1,1))
   //SYSUT3   DD  UNIT=3390,SPACE=(CYL,(1,1))
   //SYSLIN   DD  DSN=&&SYSGO,DISP=(,PASS),UNIT=3390,SPACE=(CYL,(1,1))
   //SYSIN    DD  DSN=NETVIEW.CNMSAMP(CNMS0001),DISP=SHR
   //*
   //LINK    EXEC PGM=HEWL,PARM='LIST,MAP,XREF,RENT',COND=(4,LT)
   //SYSPRINT DD  SYSOUT=*
   //SYSUT1   DD  SPACE=(CYL,(1,1)),DISP=(NEW,PASS),UNIT=3390
   //SYSLMOD  DD  DSN=MSOPS.CUSTOM.VTAMLIB(AMODETAB),DISP=SHR
   //SYSLIN   DD  DSN=&&SYSGO,DISP=(OLD,DELETE)
   //
   ```

Note: If your installation already has an AMODTAB defined but is missing the table entry for DSIL6MOD; the following MODEENT statements must be added to it:

```
***********************************************************************
*                                                                     *
*   LOGMODE ENTRY FOR 6.2 APPLICATIONS                                *
*                                                                     *
***********************************************************************
DSIL6MOD MODEENT LOGMODE=DSIL6MOD,FMPROF=X'13',TSPROF=X'07',           X
               PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'50B1',TYPE=X'00', X
               SSNDPAC=X'00',SRCVPAC=X'03',PSNDPAC=X'03',              X
               RUSIZES=X'8888',PSERVIC=X'060200000000000000002C00'
```

4. Authorize the new user defined VTAMLIB data set. Add it at the front of the VTAMLIB DD definition statements in the procedure used to start VTAM.

Step 11: Increasing the number of entries in the REXX Environment Table

This step is optional depending on whether prior installation actions have been taken. If the installation is currently running NetView as part of the normal process, then the actions described here will most likely have been done. If not, the number of entries in the REXX Environment Table (IRXANCHR) will need to be checked and increased if necessary. The following steps can be used to determine the present value that is set and increase it if necessary. For more information, see "Modifying the maximum number of language processor (REXX) environments for msys for Operations" on page 37.

1. Browse SYS1.LINKLIB(IRXANCHR). On the command line, type `HEX` and hit Enter. The following will be displayed. Check the highlighted value on your display. In this example the change had already been applied—X'01F4' is a value of 500. If the value in your display is X'0190' (a value of 400) or more you are done. Otherwise proceed to the next step to increase the number of table entries.

```
BROWSE    SYS1.LINKLIB(IRXANCHR)                      Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> CSR
********************************* Top of Data **********************************

 ------------------------------------------------------------------------------
.0......IRXANCHR......+
28000001CDECDCCD00000044
000001009971538900060E0
 ------------------------------------------------------------------------------
0³..............................................................................
8F000000000000000000000000000000000000000000000000000000000000000000000000000000
0A1000000000000000000000000000000000000000000000000000000000000000000000000000000
 ------------------------------------------------------------------------------
0..5695DF108 ......Ä.?
810FFFFCCFFF4000010636
05256954610801403F133F
 ------------------------------------------------------------------------------
0.d0..569623400 .....
818800FFFFFFFFF400001
0440105696234000130 3F
 ------------------------------------------------------------------------------
............ .+ ..+
000000000000040440044
D0000400600000E001E0
 ------------------------------------------------------------------------------
IRXANCHR0100...4................................................................
CDECDCCDFFFF000F00000002000000000000000000000000000000000000000000000000000000000
99715389010000140000000800000000000000000000000000000000000000000000000000000000
 ------------------------------------------------------------------------------
******************************** Bottom of Data ********************************
```

2. Apply the following SMP/E USERMOD to increase the number of table entries. The sample, IRXTSMPE, can be found in SYS1.SAMPLIB. Determine the FMID of MOD entry IRXANCHR from SMP/E. For z/OS 1.2 this is HTE26D2. Make

the highlighted changes, apply the USERMOD and either re-IPL or issue a 'F LLA,REFRESH' command after copying the updated IRXANCHR into the current SYS1.LINKLIB.

Note: Do not attempt to use AMASPZAP to make these changes. The number of table entries coded determines the size of this load module and incorrectly changing it will result in a system that you will not be able to log on to.

```
++ USERMOD (TSOANCH)
 /*******************************************************************
  USERMOD TO CHANGE THE NUMBER OF ENTRIES IN THE IRXANCHR TABLE TO
  SUPPORT A LARGER NUMBER OF CONCURRENT ENVIRONMENTS.
 *******************************************************************/.
++ VER (Z038) FMID(HTE26D2)     /* REPLACE XXXXXXX WITH YOUR CURRENT
                                 SYSTEM FMID              @YA60165*/.
++ SRC (IRXANCHR) SYSLIB(SAMPLIB) DISTLIB(ASAMPLIB).
       TITLE 'IRXANCHR - THE REXX ENVIRONMENT TABLE'
       MACRO
       IRXANCHR &ENTRYNUM=40
*/***START OF SPECIFICATIONS******************************************/
*/*                                                                  */
*/*  MACRO-NAME = IRXANCHR                                           */
*/*                                                                  */
*/*  COPYRIGHT =                                                     */
*/*    5685-085 COPYRIGHT IBM CORP. 1991                             */
*/*    THIS PRODUCT CONTAINS RESTRICTED MATERIALS OF IBM,            */
*/*    REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083.      */
*/*                                                                  */
*/*  DESCRIPTIVE-NAME = Macro to build IRXANCHR                      */
*/*                                                                  */
*/*  FUNCTION = IRXANCHR is a macro to be used by an installation    */
*/*             to create the REXX Environment Table. If an         */

*/*             installation decides that the default number of      */
*/*             permitted REXX environments is too small (or too     */
*/*             large) they can update it via this macro.           */
*/*                                                                  */
*/*  INSTALLATION =                                                  */
*/*             This macro as shipped in SYS1.SAMPLIB will create    */
*/*             (via SMP/E) a new IRXANCHR load module with 40       */
*/*             REXX environments. To change the number of allowable*/
*/*             environments, you must:                              */
*/*                                                                  */
*/*                  1. change the ENTRYNUM= parameter on the        */
*/*                     IRXANCHR macro invocation at the end of this */
*/*                     sample to the desired value (default is 40.) */
*/*                                                                  */
*/*                  2. change the FMID in the ++VER line to the     */
*/*                     FMID of your current TSO/E release.         */
*/*                                                                  */
*/*                  3. install following the instructions for SMP/E */
*/*                     user modifications.                         */
*/*                                                                  */
*/*  INVOCATION = MACRO SPECIFICATION IS:                            */
*/*                                                                  */
*/*               IRXANCHR ENTRYNUM=nn                               */
*/*                                                                  */
*/*               ENTRYNUM=nn specifies the number                   */
*/*               of elements for the array.                        */
*/*                                                                  */
*/*               ENTRYNUM=40 is the default.                       */
```

Figure 63. SMP/E USERMOD to increase the number of table entries (Part 1 of 2)

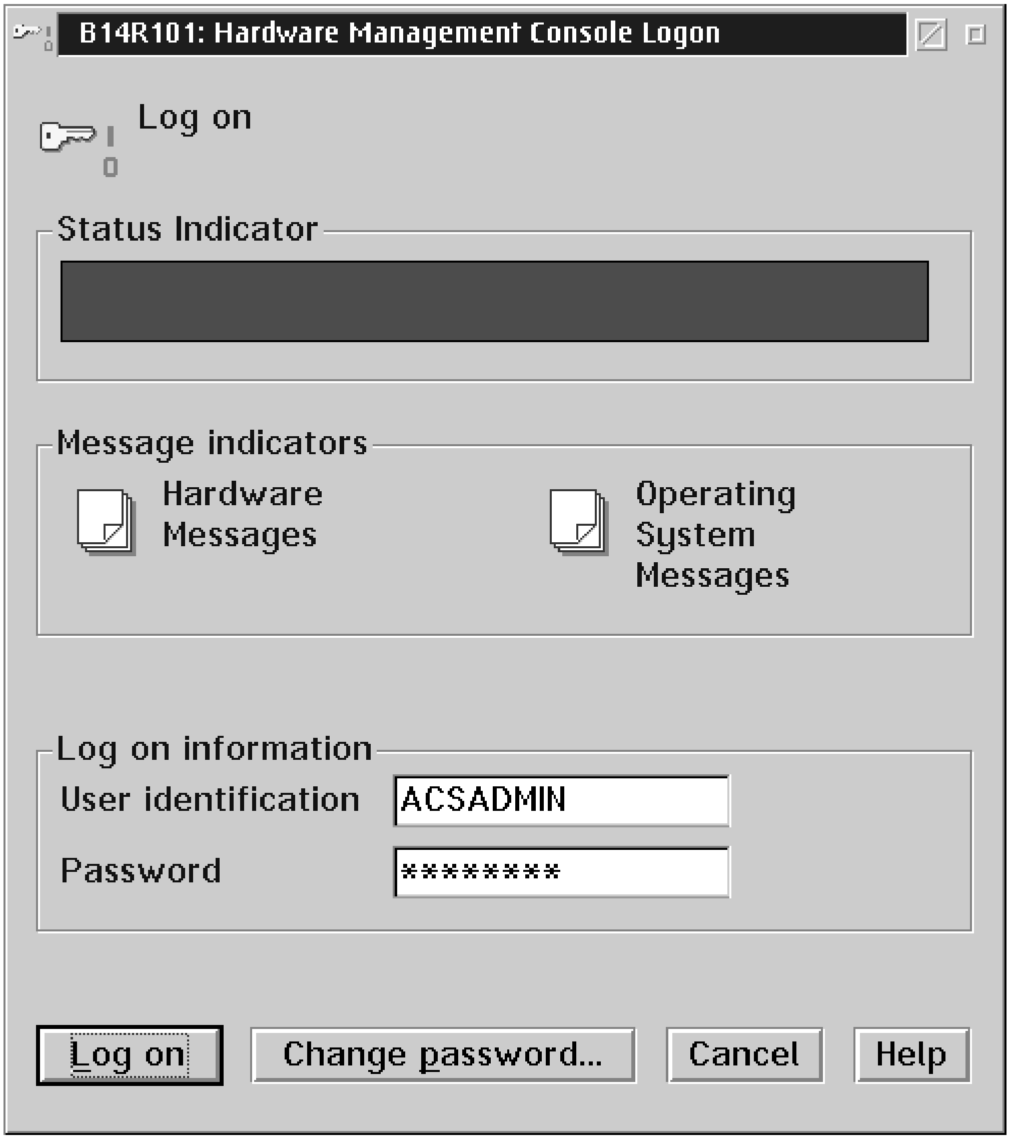
```
*/*                                                                  */
*/*  CHANGE ACTIVITY =                                               */
*/*                                                                  */
*/*        OY36194 -  Created for TSO/E Version 2 Release 1  @YA36194*/
*/*                                                                  */
*/*        OY51911 -  Versioned into JTE23X2. FMID on ++VER         */
*/*                   statement updated to JTE23X2           @YA51911*/
*/*                                                                  */
*/*        OY60165 -  Changed the FMID to XXXXXXX which the          */
*/*                   installation will replace with their           */
*/*                   current TSO/E FMID.                   @YA60165*/
*/*                                                                  */
*/***END OF SPECIFICATIONS*******************************************/

&ID      SETC  'IRXANCHR'             eye catcher
&VERSION SETC  '0100'                 version number
&TOTAL   SETA  &ENTRYNUM              number of entries in table
&LENGTH  SETA  40                     size of each entry
&ID      CSECT                        this is a load module
&ID      AMODE 31                     AMODE = 31 bit addressing
&ID      RMODE ANY                    RMODE = anywhere
ID       DC   CL8'&ID'                insert eye catcher
VERSION  DC   CL4'&VERSION'           insert version number
TOTAL    DC   F'&TOTAL'               total number of entries
USED     DC   F'0'                    number of used entries (0)
LENGTH   DC   F'&LENGTH'              length of each entry
        DC   XL8'0'                  RESERVED
FIRST    DS   0D                      first entry: double word boundry
        DC   (&TOTAL)XL&LENGTH'0'    total entries
        MEND                         end of macro
*/*******************************************************************/
*/*                                                                  */
*/*  IRXANCHR - The REXX environment table                           */
*/*                                                                  */
*/*  To change the number of allowable REXX environments, you must:  */
*/*                                                                  */
*/*                                                                  */
*/*                  1. change the ENTRYNUM= parameter on the        */
*/*                     IRXANCHR macro invocation to the desired     */
*/*                     value (default is 40.)                       */
*/*                                                                  */
*/*                  2. change the FMID in the ++VER line to the     */
*/*                     FMID of your current TSO/E release.         */
*/*                                                                  */
*/*                  3. install following the instructions for SMP/E */
*/*                     user modifications.                         */
*/*                                                                  */
*/*******************************************************************/
        IRXANCHR ENTRYNUM=400
       END
```

Figure 63. SMP/E USERMOD to increase the number of table entries (Part 2 of 2)

Step 12: Hardware customization of the SEs

This step describes the hardware customization that must be performed on every Support Element (SE) that was defined in the AOFCUST policy in Step 9: Customizing the member AOFCUST. Carrying out this step will require you to reboot the Support Element. For more information, see “Preparing the Support Element” on page 32.

1. Logon as ACSADMIN. If you do this from the change management HMC, every CPC on the HMC/SE LAN can be reached. These changes must be performed on every CPC where msys for Operations hardware control is desired.


B14R101: Hardware Management Console Logon

Log on

Status Indicator

Message indicators

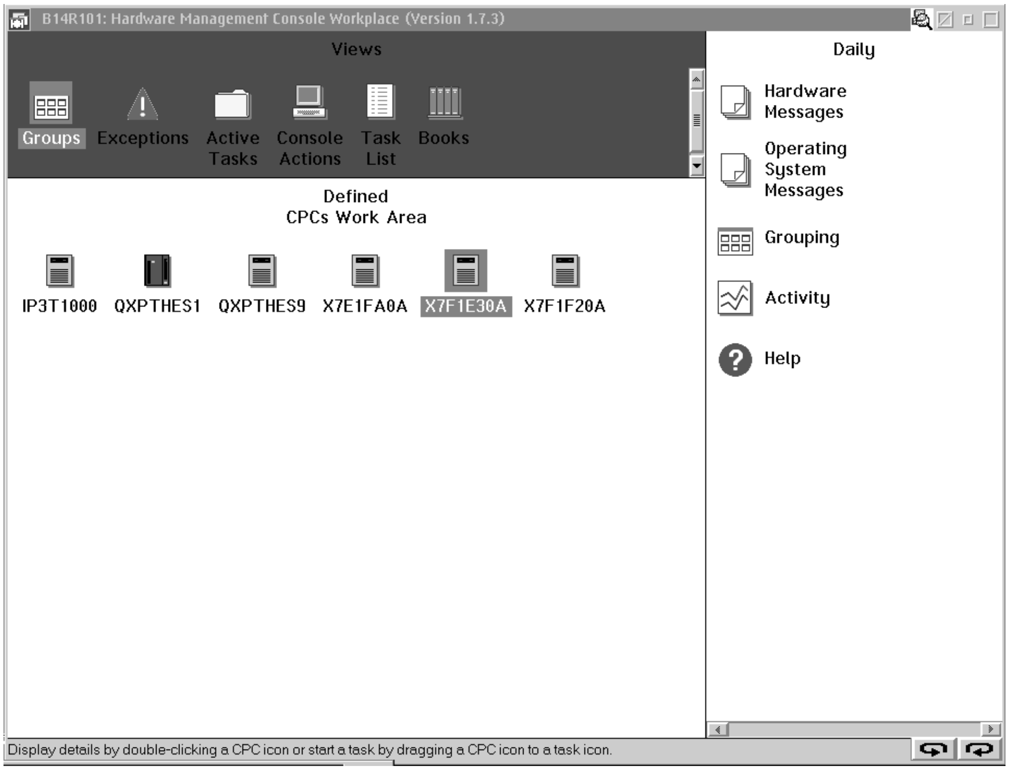
Log on information

User identification ACSADMIN

Password *********

Figure 64. Log on as ACSADMIN

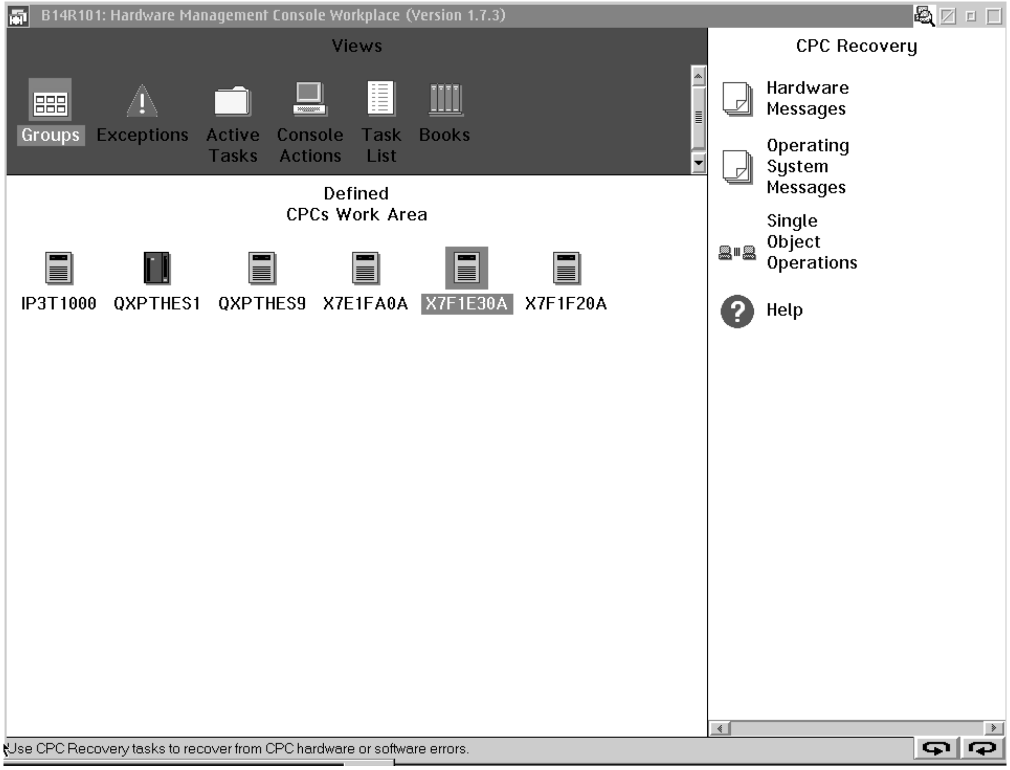
2. Select the **Groups** icon in the *Views* window and double click on the **Defined CPCs** icon.



Display details by double-clicking a CPC icon or start a task by dragging a CPC icon to a task icon.

Figure 65. Select the Groups icon

3. Select the CPC that you are changing and drag it to, or double click on, the **Single Object Operations** icon.

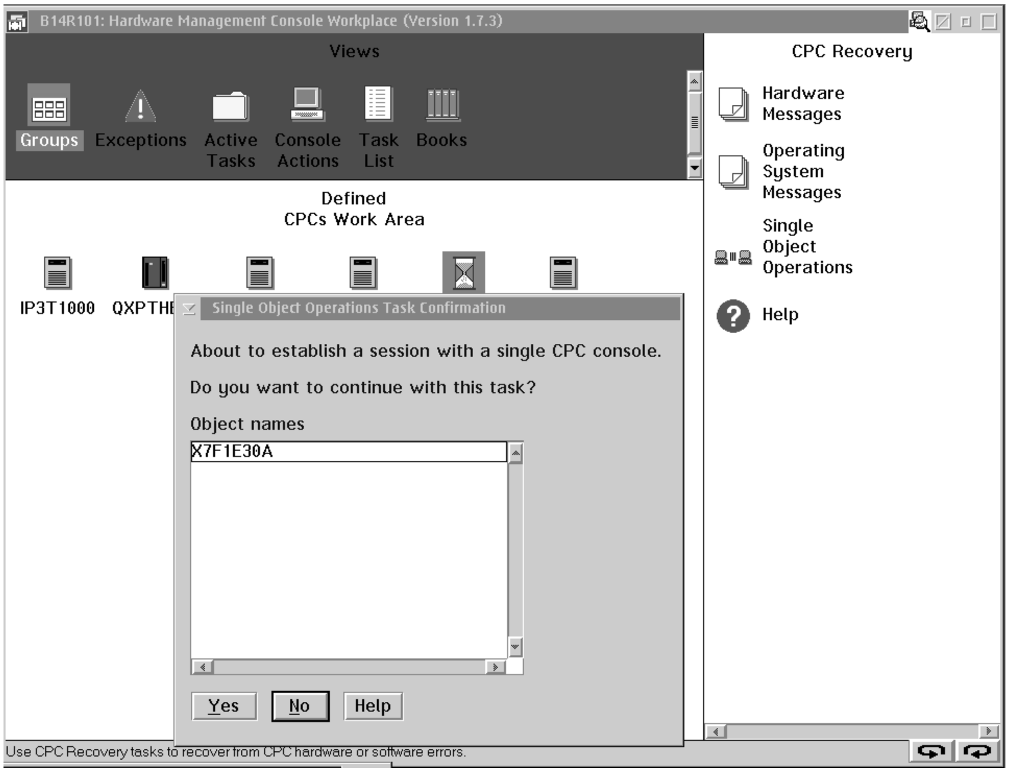


B14R101: Hardware Management Console Workplace (Version 1.7.3)

Use CPC Recovery tasks to recover from CPC hardware or software errors.

Figure 66. Select the required CPC

This will establish a session with the SE for that CPC; select **Yes** to establish the session. Any changes made here can also be performed directly at the SE by logging on there. However, when making changes to multiple CPCs, the HMC is more convenient.



About to establish a session with a single CPC console.

Do you want to continue with this task?

Object names

X7F1E30A

Use CPC Recovery tasks to recover from CPC hardware or software errors.

Figure 67. Single Object Operations Task Confirmation panel

4. Double click the **Support Element Settings** icon in the *Console Actions Work Area*.

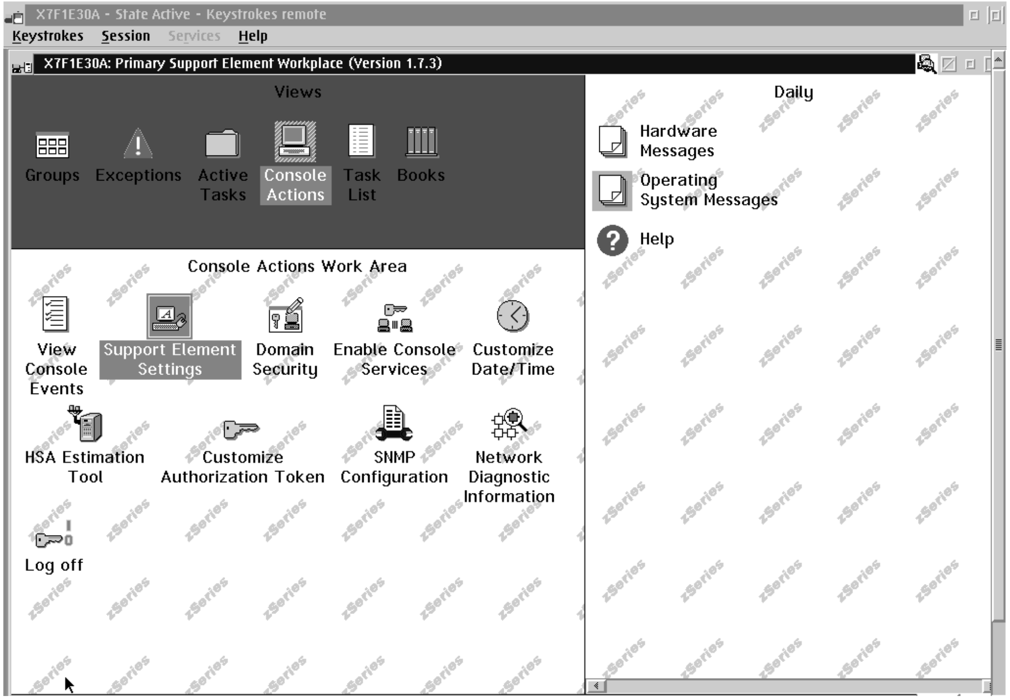
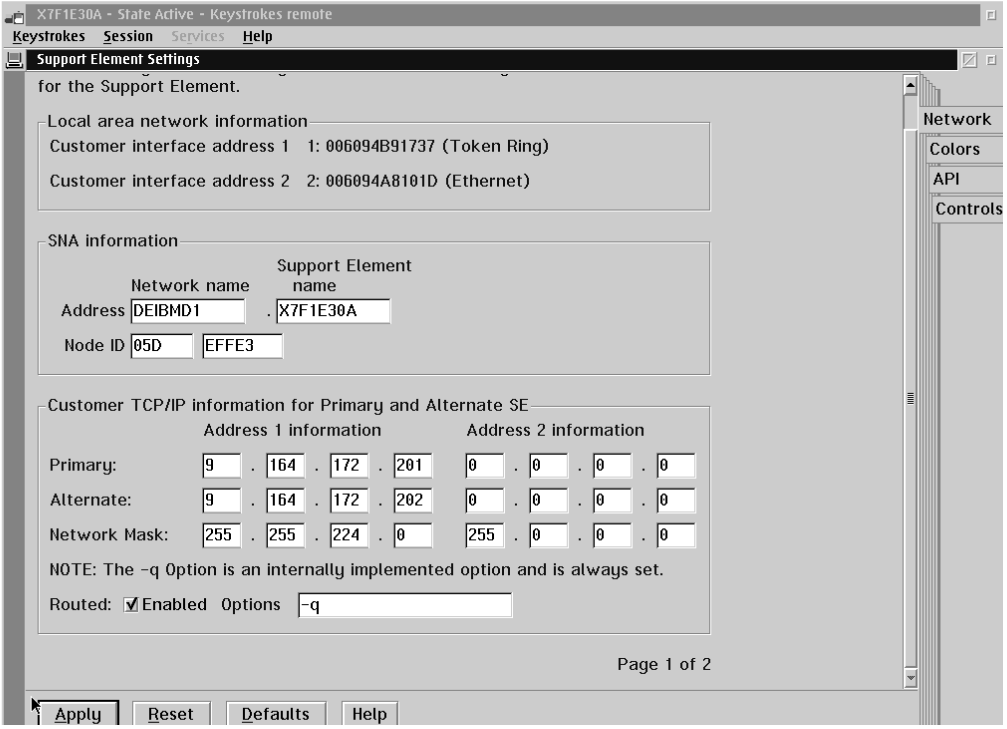


Figure 68. Select the Support Element Settings icon

5. Select the **Network** tab, if it is not already visible. Make a note of the Primary SE IP Address because you will need it when creating the 1st community name in 8 on page 319.



Support Element Settings

for the Support Element.

Local area network information

Customer interface address 1 1: 006094B91737 (Token Ring)

Customer interface address 2 2: 006094A8101D (Ethernet)

SNA information

	Network name		Support Element name
Address	DEIBMD1	.	X7F1E30A
Node ID	05D		EFFE3

Customer TCP/IP information for Primary and Alternate SE

	Address 1 information	Address 2 information
Primary:	9 . 164 . 172 . 201	0 . 0 . 0 . 0
Alternate:	9 . 164 . 172 . 202	0 . 0 . 0 . 0
Network Mask:	255 . 255 . 224 . 0	255 . 0 . 0 . 0

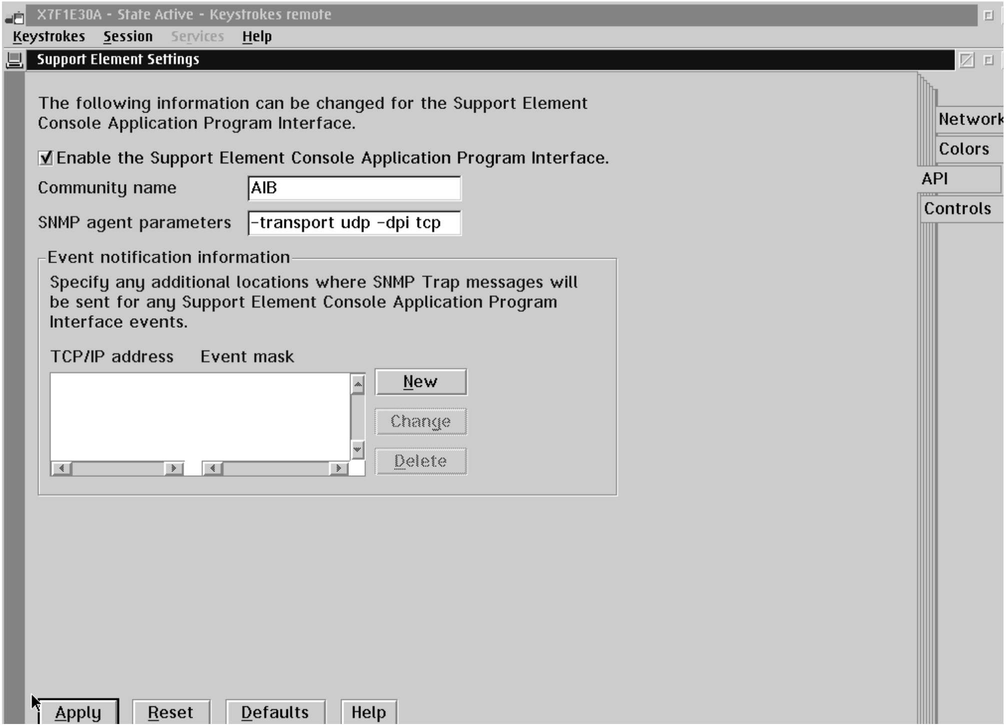
NOTE: The -q Option is an internally implemented option and is always set.

Routed: ☑ Enabled Options -q

Page 1 of 2

Figure 69. Network tab in the Support Element Settings notebook window

6. Select the **API** tab. Make sure that the *Enable the Support Element Console Application Program Interface* check box is selected. Fill in the **Community Name** field. The value coded here can be anything you choose *except* the value coded in the HW Section of AOFCUST on the CPC statement. This field is case sensitive. Use *upper case* and insert or overwrite any existing value. The **SNMP agent parameters** field should already be set. If not, set it to `-transport udp -dpi tcp`.



Support Element Settings

The following information can be changed for the Support Element Console Application Program Interface.

☑ Enable the Support Element Console Application Program Interface.

Community name: AIB

SNMP agent parameters: -transport udp -dpi tcp

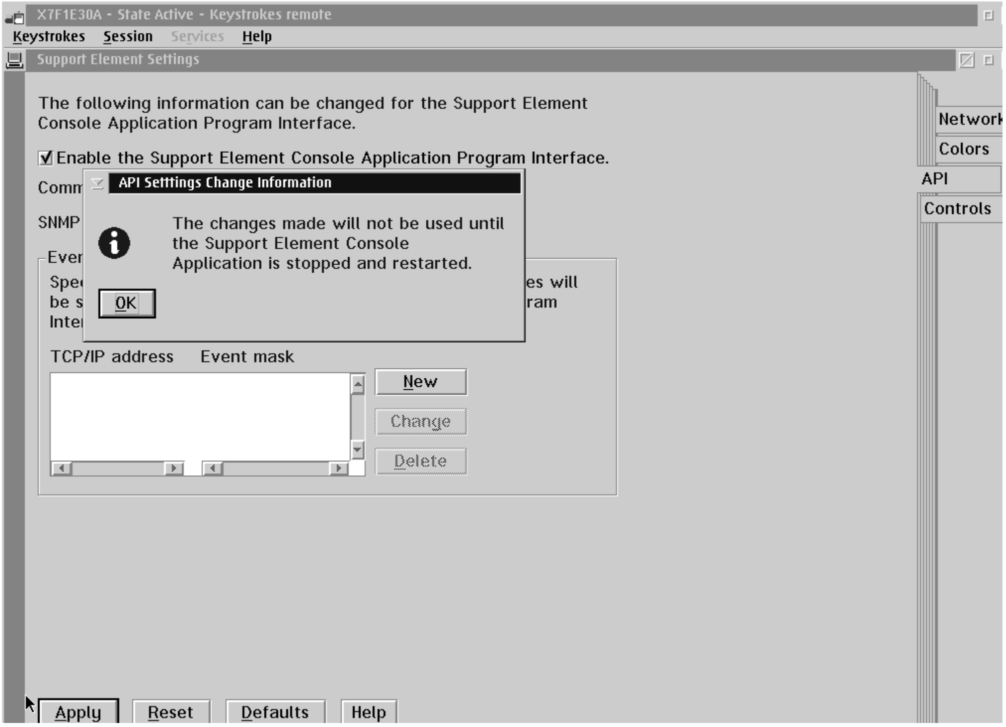
Event notification information

Specify any additional locations where SNMP Trap messages will be sent for any Support Element Console Application Program Interface events.

TCP/IP address   Event mask

Figure 70. API tab in the Support Element Settings notebook window

Select the **Apply** push button to save the changes. A panel informs you that changes on this panel require you to reboot the SE - select the **OK** button to continue. You can reboot the SE when setup is complete.



Support Element Settings

The following information can be changed for the Support Element Console Application Program Interface.

☑ Enable the Support Element Console Application Program Interface.

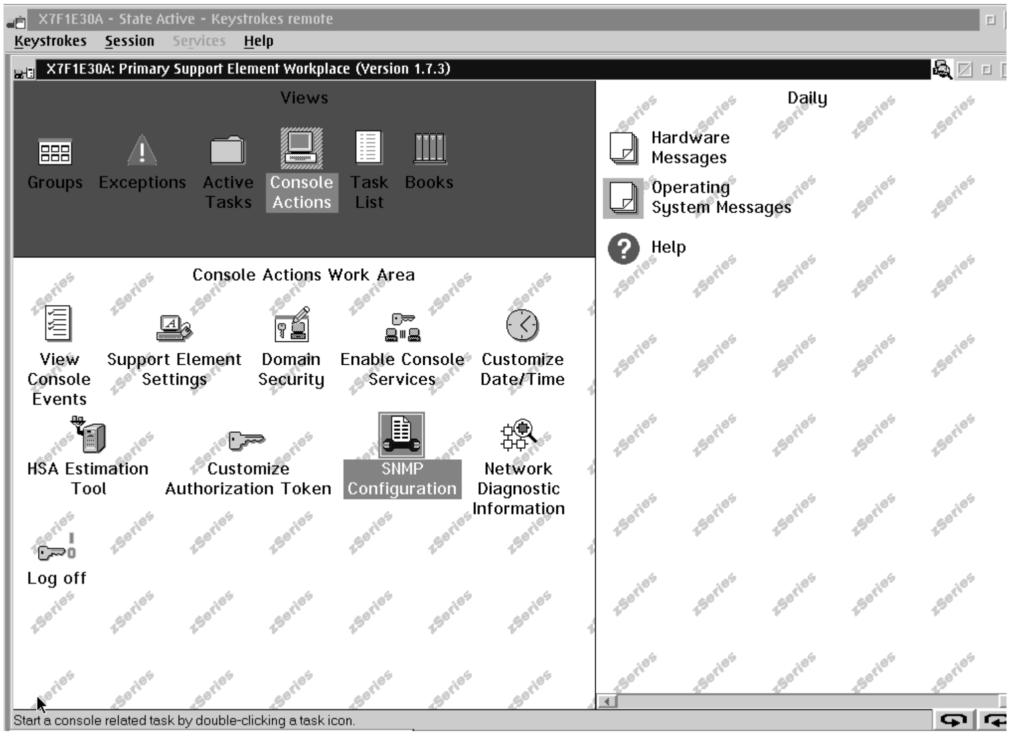
API Setttings Change Information

The changes made will not be used until the Support Element Console Application is stopped and restarted.

TCP/IP address   Event mask

Figure 71. API Settings Change Information panel

7. Double click the **SNMP Configuration** icon in the *Console Actions* view.



Start a console related task by double-clicking a task icon.

Figure 72. Select the SNMP Configuration icon

8. If not already active, select the **Communities** tab. Two community names need to be created on this panel, as shown below. The *Name* field is case sensitive. Use *upper case*.

   a. Create an entry identical to the name specified in *Support Element Settings* in 6. Enter the following information and select the **Add** push button to add the new community name:

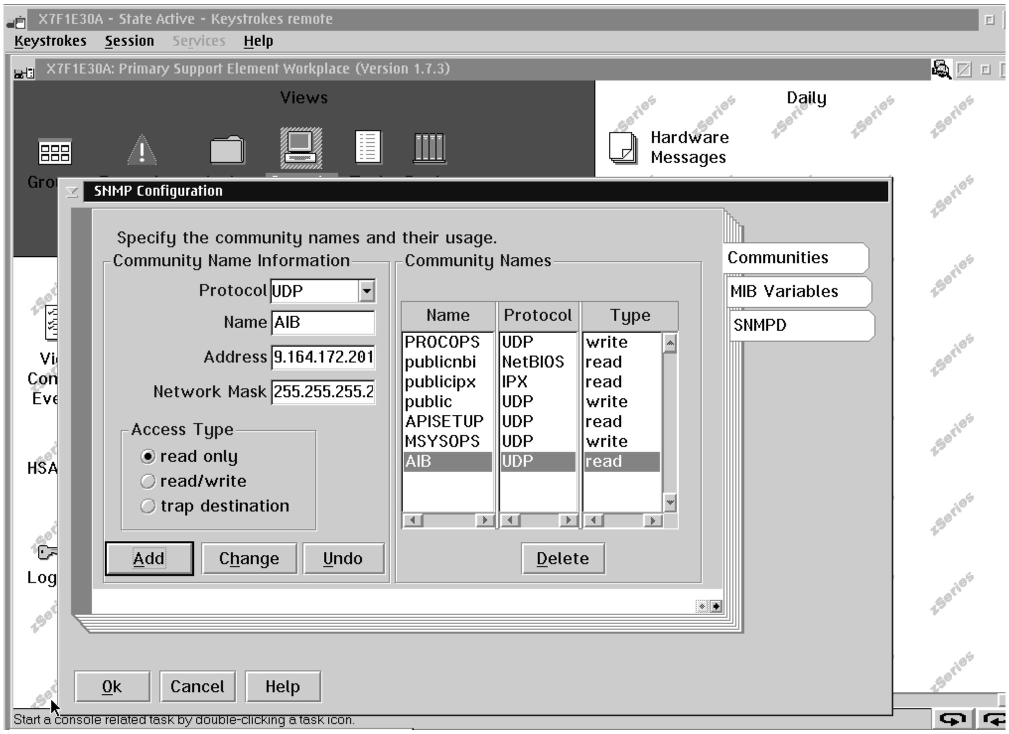
      Protocol
      Select UDP from the drop-down list

      Name
      The API Community name you have chosen, in this case AIB

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

      Network Mask
      This value must be 255.255.255.255

      Access Type
      Select the **read only** radio button



Specify the community names and their usage.

Community Name Information

Protocol UDP

Name AIB

Address 9.164.172.201

Network Mask 255.255.255.2

Access Type

● read only

○ read/write

○ trap destination

Community Names

Name	Protocol	Type
PROCOPS	UDP	write
publicnbi	NetBIOS	read
publicipx	IPX	read
public	UDP	write
APISETUP	UDP	read
MSYSOPS	UDP	write
AIB	UDP	read

Start a console related task by double-clicking a task icon.

Figure 73. Set the first community name

b. Create an entry identical to the value coded in the HW Section of AOFCUST. Enter the following information and select the **Add** push button to add the new community name:

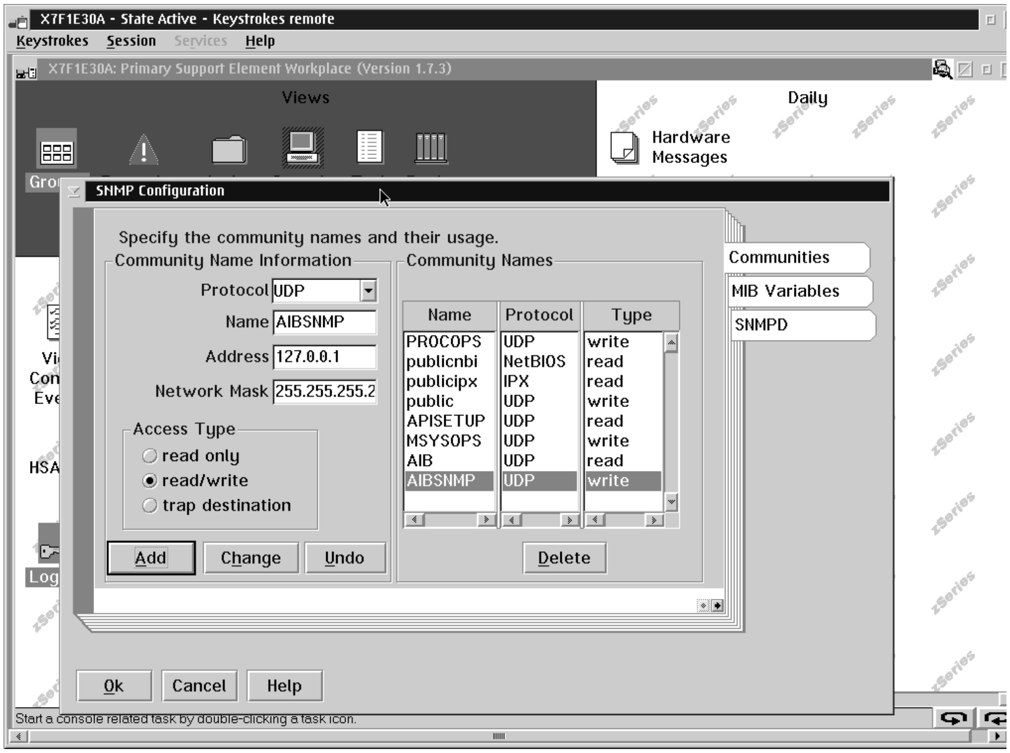
Protocol
Select UDP from the drop-down list

Name
In this case, the name is AIBSNMP

Address
This value must be 127.0.0.1

Network Mask
This value must be 255.255.255.255

Access Type
Select the **read/write** radio button


X7F1E30A - State Active - Keystrokes remote
Keystrokes
Session
Services
Help
X7F1E30A: Primary Support Element Workplace (Version 1.7.3)
Views
Daily
Hardware Messages
SNMP Configuration
Communities
MIB Variables
SNMPD
Add
Change
Undo
Delete
Ok
Cancel
Help
Start a console related task by double-clicking a task icon.

Specify the community names and their usage.

Community Name Information

Protocol: UDP

Name: AIBSNMP

Address: 127.0.0.1

Network Mask: 255.255.255.2

Access Type

○ read only

● read/write

○ trap destination

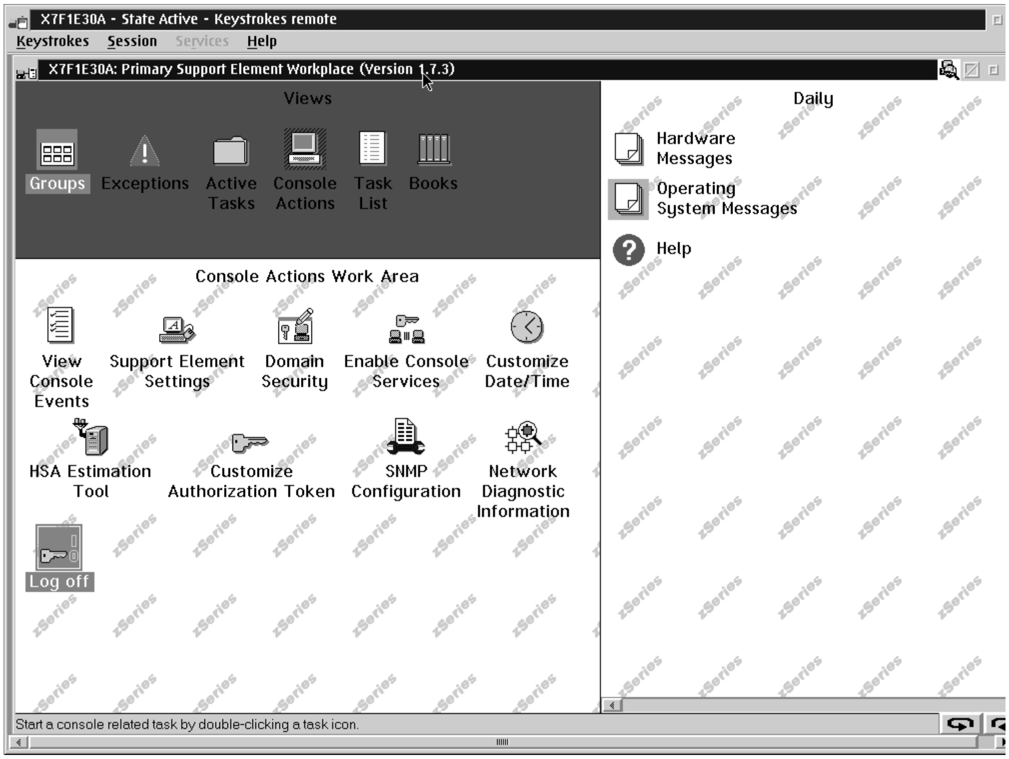
Community Names

Name	Protocol	Type
PROCOPS	UDP	write
publicnbi	NetBIOS	read
publicipx	IPX	read
public	UDP	write
APISETUP	UDP	read
MSYSOPS	UDP	write
AIB	UDP	read
AIBSNMP	UDP	write

Figure 74. Set the second community name

c. Click the **OK** button when both community names have been created.

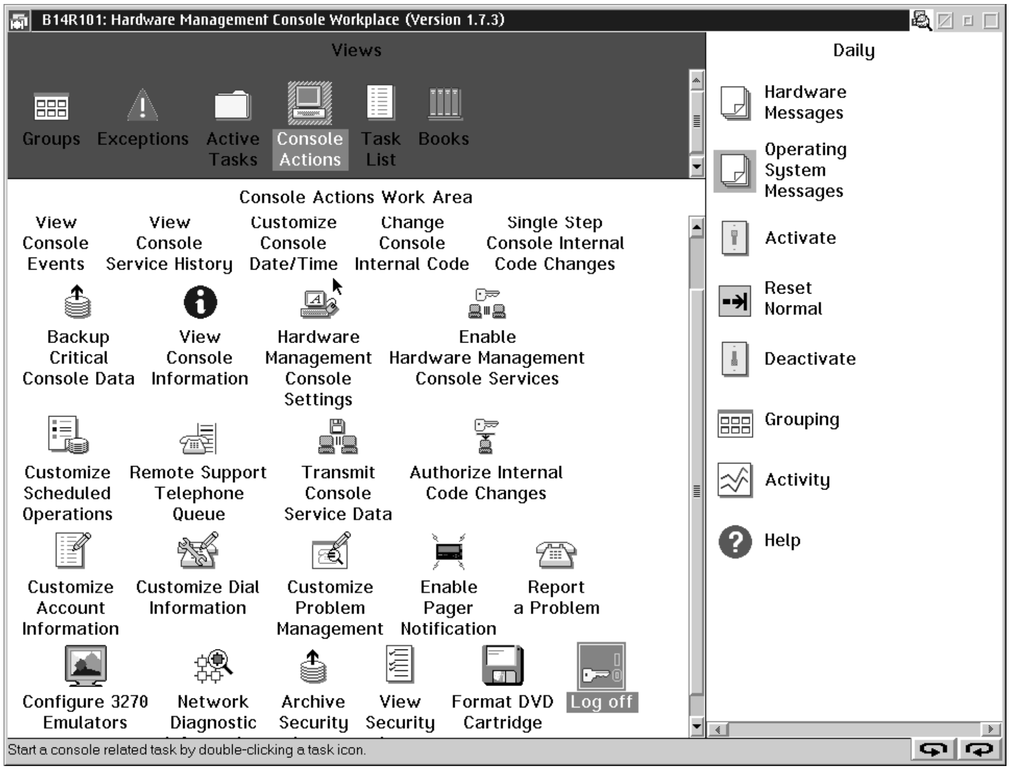
9. To end your *Single Object Operations* session, double click the **Log off** icon in the *Console Actions Work Area*.



Start a console related task by double-clicking a task icon.

Figure 75. End the Single Object Operations session

10. To end your *HMC* session as **ACSADMIN**, double click the **Logoff** icon in the *Console Actions Work Area*.



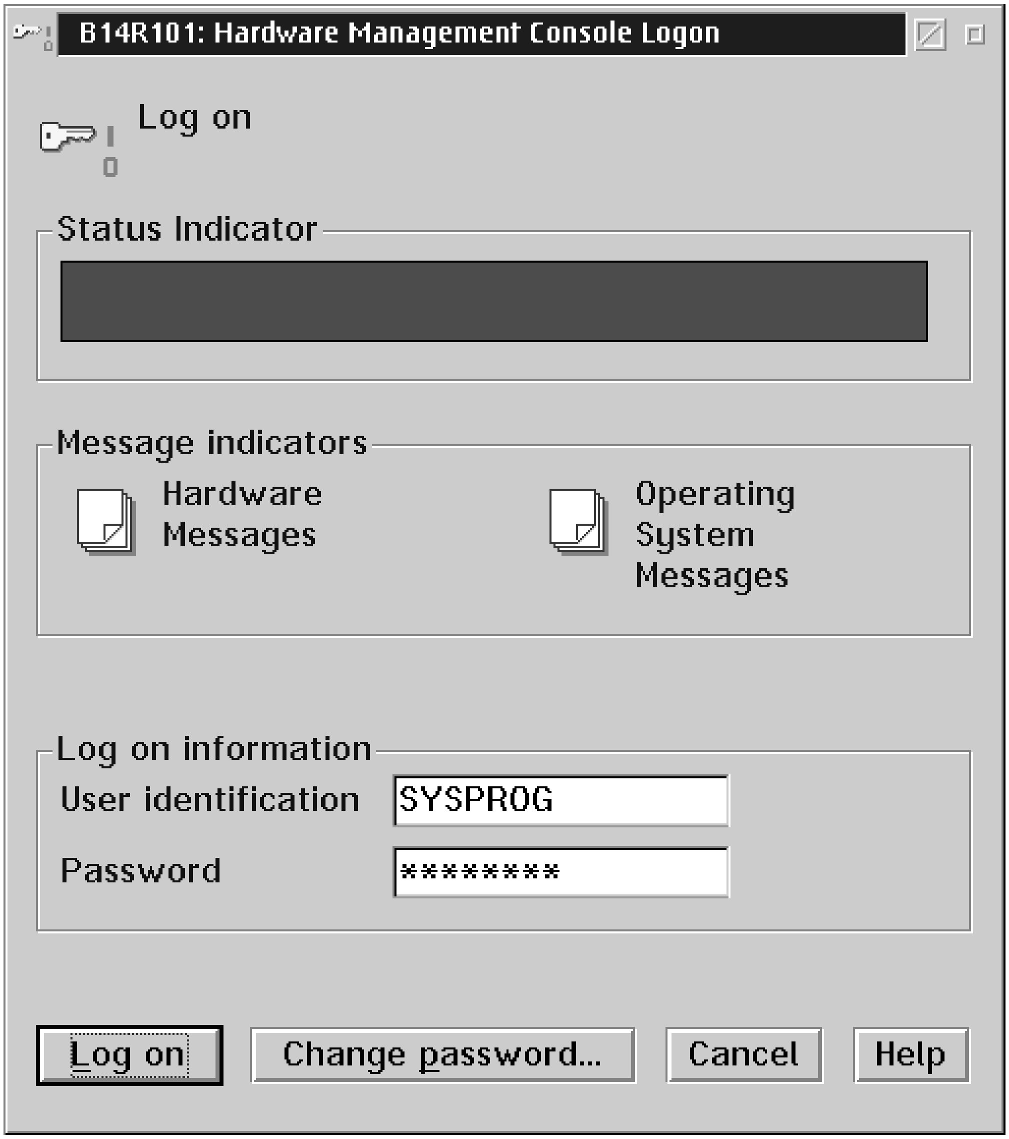
B14R101: Hardware Management Console Workplace (Version 1.7.3)

Views
Groups
Exceptions
Active Tasks
Console Actions
Task List
Books
Console Actions Work Area
View Console Events
View Console Service History
Customize Console Date/Time
Change Console Internal Code
Single Step Console Internal Code Changes
Backup Critical Console Data
View Console Information
Hardware Management Console Settings
Enable Hardware Management Console Services
Customize Scheduled Operations
Remote Support Telephone Queue
Transmit Console Service Data
Authorize Internal Code Changes
Customize Account Information
Customize Dial Information
Customize Problem Management
Enable Pager Notification
Report a Problem
Configure 3270 Emulators
Network Diagnostic
Archive Security
View Security
Format DVD Cartridge
Log off
Daily
Hardware Messages
Operating System Messages
Activate
Reset Normal
Deactivate
Grouping
Activity
Help

Start a console related task by double-clicking a task icon.

Figure 76. Log off as ACSADMIN

11. Log on as **SYSPROG**.

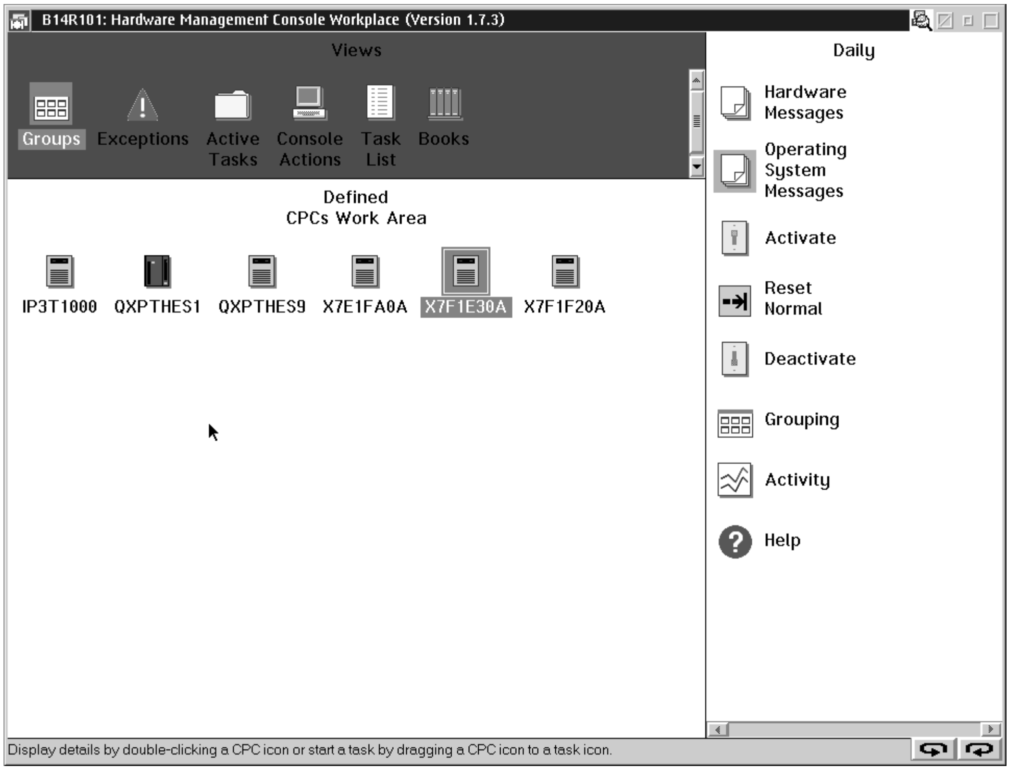


User identification SYSPROG

Password *********

Figure 77. Log on as SYSPROG

12. Select the **Groups** icon in the *Views* window and double click on the **Defined CPCs** icon.



B14R101: Hardware Management Console Workplace (Version 1.7.3)

Display details by double-clicking a CPC icon or start a task by dragging a CPC icon to a task icon.

Figure 78. Select the Groups icon

13. Select the CPC that you are changing and drag it to, or double click on, the **Single Object Operations** icon in the *CPC Recovery* task window.

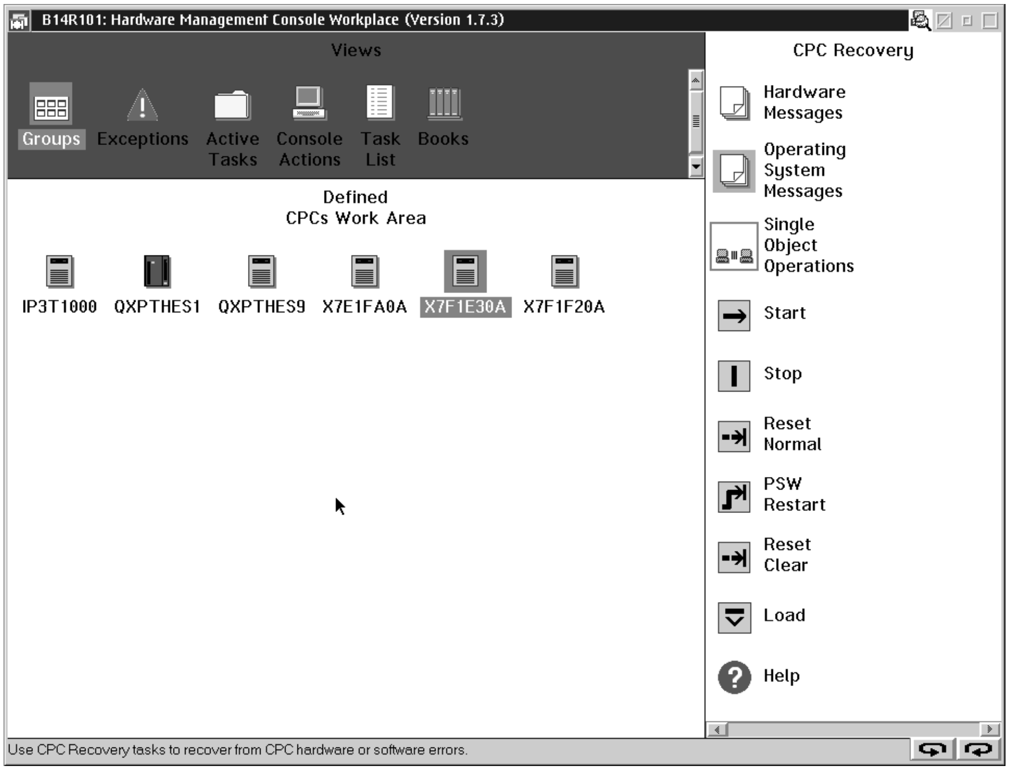
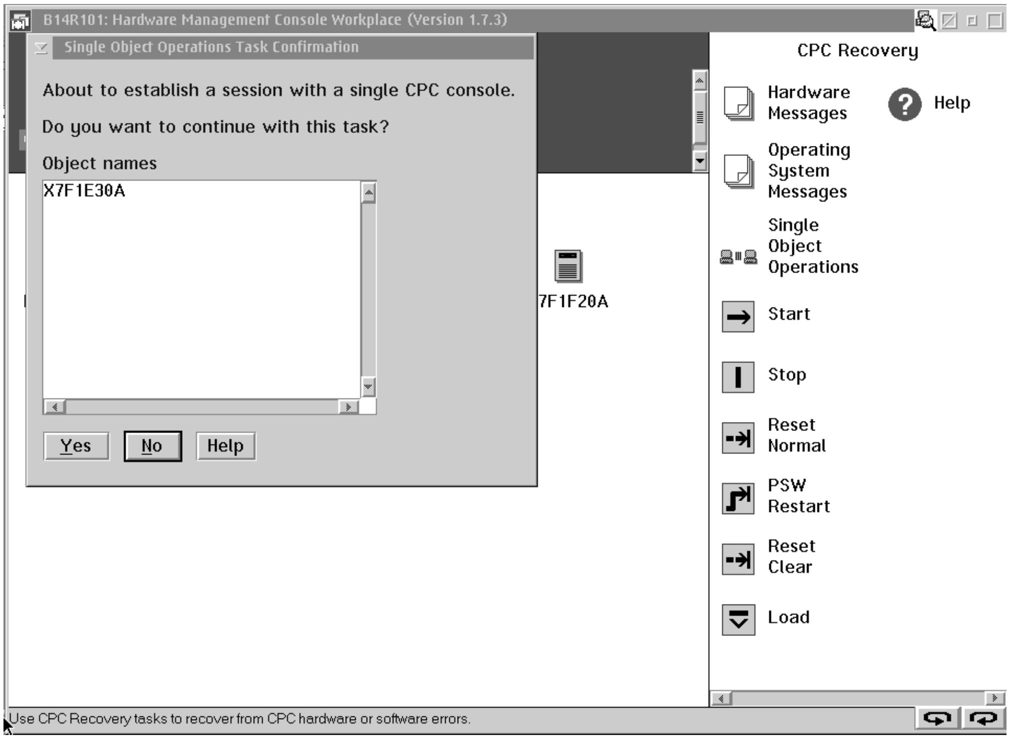


Figure 79. Select the required CPC

This will establish a session with the SE for that CPC; select **Yes** to establish the session. This takes you back into the SE for that CPC and is necessary because this stage of the setup cannot be completed under ACSADMIN.


B14R101: Hardware Management Console Workplace (Version 1.7.3)

Single Object Operations Task Confirmation

About to establish a session with a single CPC console.

Do you want to continue with this task?

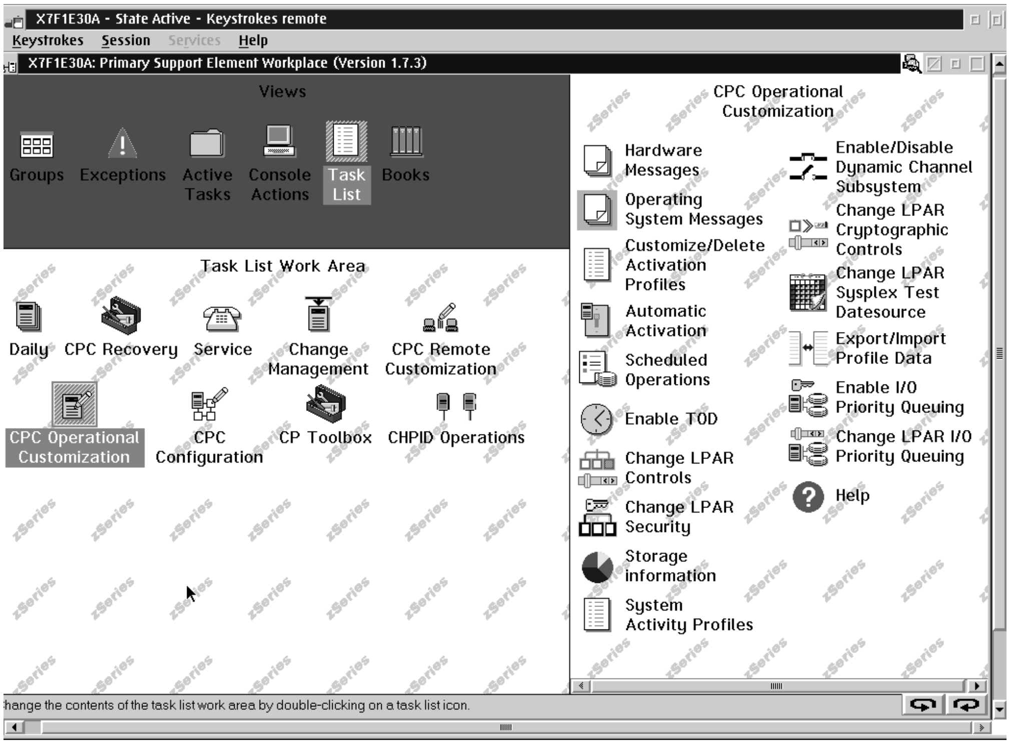
Object names

X7F1E30A

Use CPC Recovery tasks to recover from CPC hardware or software errors.

Figure 80. Single Object Operations Task Confirmation panel

14. Select the **Task List** icon in the *Views* area. Cycle the task window until the *CPC Operational Customization* icon is displayed, or select it from the *Task List Work Area* view.

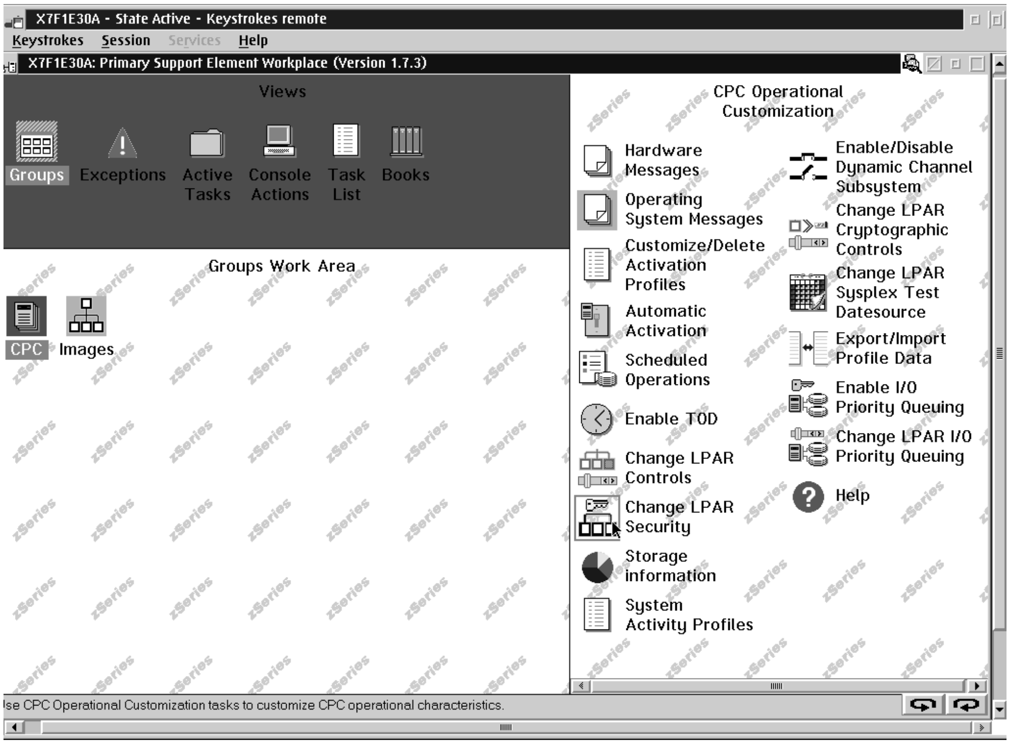

X7F1E30A - State Active - Keystrokes remote

X7F1E30A: Primary Support Element Workplace (Version 1.7.3)

hange the contents of the task list work area by double-clicking on a task list icon.

Figure 81. Select the CPC Operational Customization icon

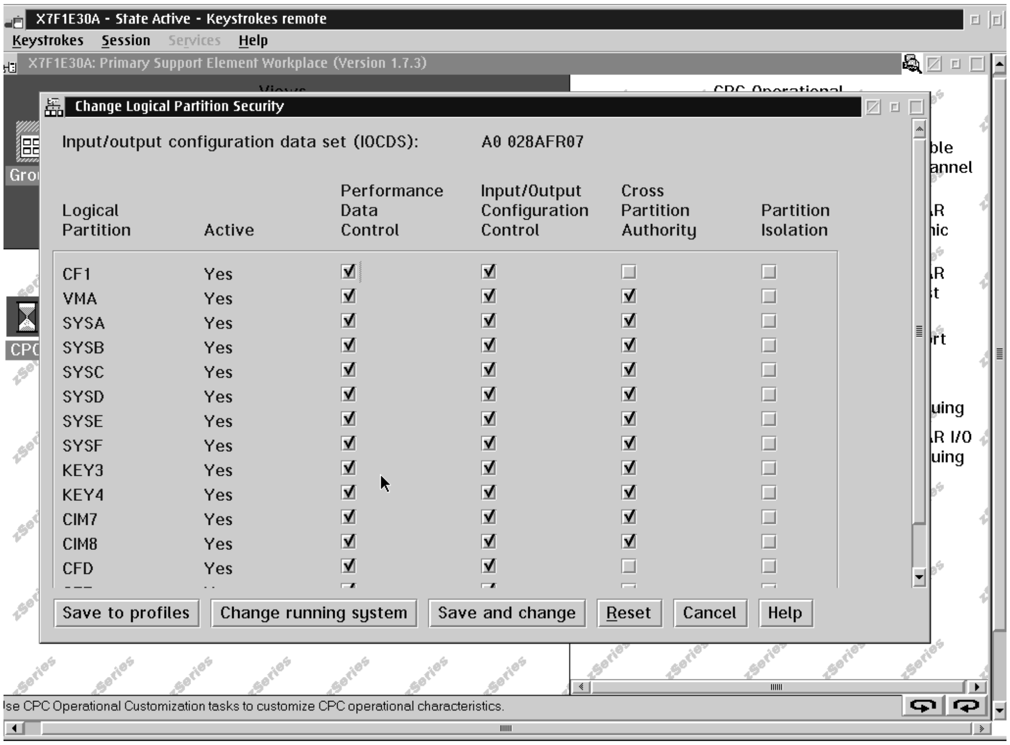
15. Select the **Groups** icon in the *Views* area and double click the **Change LPAR Security** icon in the *CPC Operational Customization* task window.


X7F1E30A - State Active - Keystrokes remote
Keystrokes
Session
Services
Help
X7F1E30A: Primary Support Element Workplace (Version 1.7.3)
Views
Groups
Exceptions
Active Tasks
Console Actions
Task List
Books
Groups Work Area
CPC
Images
CPC Operational Customization
Hardware Messages
Operating System Messages
Customize/Delete Activation Profiles
Automatic Activation
Scheduled Operations
Enable TOD
Change LPAR Controls
Change LPAR Security
Storage information
System Activity Profiles
Enable/Disable Dynamic Channel Subsystem
Change LPAR Cryptographic Controls
Change LPAR Sysplex Test Datesource
Export/Import Profile Data
Enable I/O Priority Queuing
Change LPAR I/O Priority Queuing
Help

Ise CPC Operational Customization tasks to customize CPC operational characteristics.

Figure 82. Select the Change LPAR Security icon

16. Click the check boxes under *Cross Partition Authority* for those Logical Partitions on which the Internal Hardware Transport is to be enabled. Click the **Save and change** button.


X7F1E30A - State Active - Keystrokes remote
Keystrokes
Session
Services
Help
X7F1E30A: Primary Support Element Workplace (Version 1.7.3)
Change Logical Partition Security
Save to profiles
Change running system
Save and change
Reset
Cancel
Help

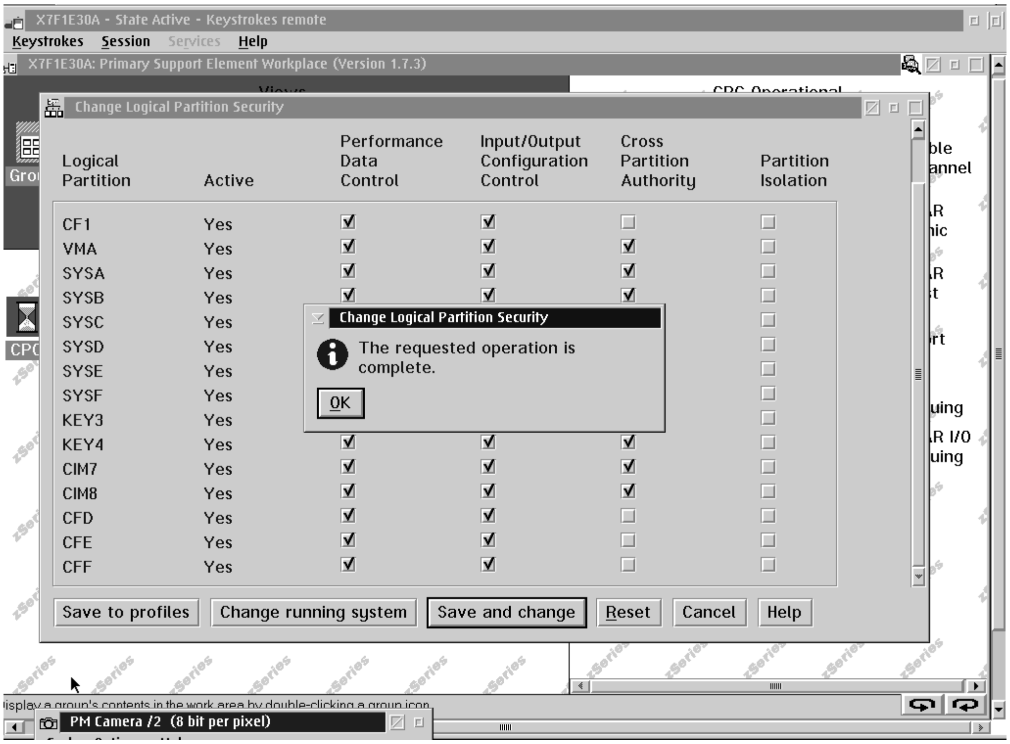
Input/output configuration data set (IOCDS): A0 028AFR07

Logical Partition	Active	Performance Data Control	Input/Output Configuration Control	Cross Partition Authority	Partition Isolation
CF1	Yes	☑	☑	☐	☐
VMA	Yes	☑	☑	☑	☐
SYSA	Yes	☑	☑	☑	☐
SYSB	Yes	☑	☑	☑	☐
SYSC	Yes	☑	☑	☑	☐
SYSD	Yes	☑	☑	☑	☐
SYSE	Yes	☑	☑	☑	☐
SYSF	Yes	☑	☑	☑	☐
KEY3	Yes	☑	☑	☑	☐
KEY4	Yes	☑	☑	☑	☐
CIM7	Yes	☑	☑	☑	☐
CIM8	Yes	☑	☑	☑	☐
CFD	Yes	☑	☑	☐	☐

Use CPC Operational Customization tasks to customize CPC operational characteristics.

Figure 83. Set the Cross Partition Authority

17. Click the **OK** button on the panel that informs you that the operation is complete.


X7F1E30A - State Active - Keystrokes remote
Keystrokes
Session
Services
Help
X7F1E30A: Primary Support Element Workplace (Version 1.7.3)
Change Logical Partition Security
Change Logical Partition Security
OK
Save to profiles
Change running system
Save and change
Reset
Cancel
Help
PM Camera /2 (8 bit per pixel)

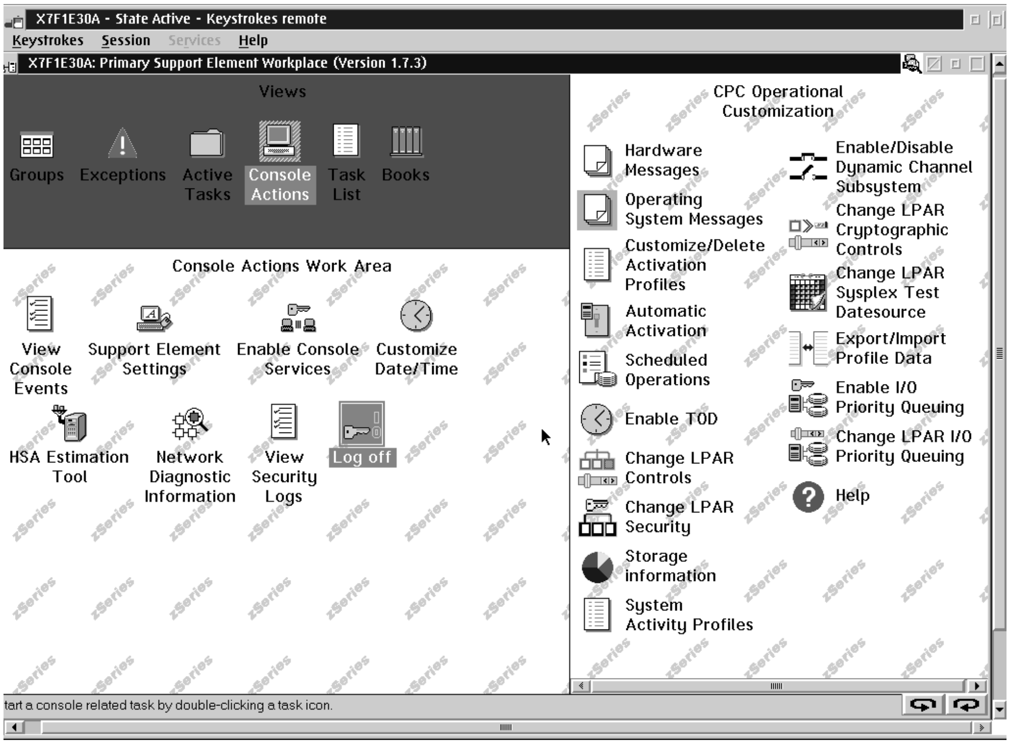
Logical Partition	Active	Performance Data Control	Input/Output Configuration Control	Cross Partition Authority	Partition Isolation
CF1	Yes	☑	☑	☐	☐
VMA	Yes	☑	☑	☑	☐
SYSA	Yes	☑	☑	☑	☐
SYSB	Yes	☑	☑	☑	☐
SYSC	Yes				☐
SYSD	Yes				☐
SYSE	Yes				☐
SYSF	Yes				☐
KEY3	Yes				☐
KEY4	Yes	☑	☑	☑	☐
CIM7	Yes	☑	☑	☑	☐
CIM8	Yes	☑	☑	☑	☐
CFD	Yes	☑	☑	☐	☐
CFE	Yes	☑	☑	☐	☐
CFF	Yes	☑	☑	☐	☐

The requested operation is complete.

Display a group's contents in the work area by double-clicking a group icon.

Figure 84. Change Logical Partition Security notification panel

18. To end your *Single Object Operations* session, select the **Console Actions** icon and double click the **Logoff** icon in the *Console Actions Work Area*.



X7F1E30A - State Active - Keystrokes remote

X7F1E30A: Primary Support Element Workplace (Version 1.7.3)

tart a console related task by double-clicking a task icon.

Figure 85. End the Single Object Operations session

19. This completes the setup for a single CPC and must be repeated for every CPC that will be under the msys for Operations sphere of control. Repeat steps 1 through 18 for the remaining CPCs that require this customization. Once this has been, done you must reboot the Support Element for the changes that you have made to it to become active.

Appendix E. The IBM Health Checker for z/OS and Sysplex checks

This appendix provides details of the checks carried out by the IBM Health Checker for z/OS and Sysplex.

Table 17 gives a list of the check names and indicates whether they are local or global, and what their interval is for repetitive checks:

Table 17. Overview of HealthChecker best practices checksOverview of HealthChecker Best Practices Checks

Check name	Page	Type	Interval
ALTERNATE_CONSOLE_GROUPS	335	local	24 hours
AMRF_AND_MPF_CONSISTENT	336	local	24 hours
AVAILABLE_FRAME_QUEUE_THRESHOLDS	332	local	24 hours
CDS_DATASET_SEPARATION	333	global	1 hour
CONSOLE_MASTER	335	local	24 hours
CONSOLE_MSCOPE_AND_ROUTCODES	336	local	24 hours
CONSOLE_NAMES	335	local	24 hours
CONSOLE_ROUTCODE_11	336	local	24 hours
COUPLINGFACILITY_STRUCTURE	333	local	12 hours
EMCS_HARDCOPY	337	local	12 hours
EMCS_MSCOPE_AND_ROUTCODES	336	local	12 hours
GRS_MODE	339	global	24 hours
NUMBER_EMCS_CONSOLES	337	global	12 hours
REAL_STORAGE_AVAILABILITY	338	local	24 hours
RSU_STORAGE_AVAILABILITY	338	local	24 hours
SDUMP_AVAILABILITY	339	local	24 hours
SYS_CF_STR_REPORT	334	global	24 hours
SYSCONS_MASTER	338	local	8 hours
SYSCONS_MSCOPE	337	local	24 hours
SYSCONS_PD_MODE	337	local	24 hours
SYSCONS_ROUTCODES	337	local	24 hours
SYSPLEX_MASTER	335	global	24 hours
USS_FILESYS_CONFIG	332	local	24 hours
XCF_CLEANUP_VALUE	338	local	24 hours
XCF_FAILURE_DETECTION_INTERVAL	338	local	24 hours
XCF_SIGNALLING	334	local	12 hours
XCF_SIGNALLING_STRUCTURES_IN_CF	334	local	1 hour
XCF_SYSPLEX_FAILURE_MANAGEMENT	339	global	24 hours

Note: In the following list of checks, ″User override″ refers to your ability to specify parameters that override the IBM values. A subset of the checks support this. However, all checks can be individually disabled so that the check is not run.

- **Automove setup verification**

  Check name: USS_FILESYS_CONFIG

  Best practice: You should define your version and sysplex root HFS data as AUTOMOVE, and define your system-specific file systems as UNMOUNT. Do not define a file system as NOAUTOMOVE or UNMOUNT and a file system underneath it as AUTOMOVE. If you do, the file system defined as AUTOMOVE will not be available until the failing system is restarted. A sysplex file system that changes ownership as the result of a system failure, will only be accessible in the new environment if its mount point is also accessible. The Automove check verifies that your file systems are setup according to these rules. This check is only applicable for images that are part of a sysplex.

  The AUTOMOVE|NOAUTOMOVE|UNMOUNT parameters on ROOT and MOUNT indicate what happens to the file system if the system that owns that file system goes down. The AUTOMOVE parameter specifies that ownership of the file system is automatically moved to another system. It is the default. The NOAUTOMOVE parameter specifies that the file system will not be moved if the owning system goes down and the file system is not accessible. – UNMOUNT specifies that the file system will be unmounted when the system leaves the sysplex.

  User override: Yes

  Reference: See *z/OS UNIX System Services Planning*, GA22-7800 and APAR II3129.

- **Available frame queue threshold, reclaiming storage frames**

  Check name: Available_Frame_Queue_Thresholds

  Best practice: To avoid situations where the system does not start to reclaim storage frames soon enough, you should evaluate the values for storage. If you are running in 31-bit mode, then both the MCCAFCTH and the MCCAECTH values are used. If you are running in 64-bit mode, then only the MCCAECTH value is used. For migrations to a 64-bit environment, this check is critical because the same value used in 31-bit mode could introduce problems. IBM suggests that the IEAOPT*xx* parameters are set as follows:

  – MCCAFCTH

    MCCAFCTH specifies the low and the OK threshold values for central storage. The *lowvalue* indicates the number of frames on the available frame queue when stealing begins. The *okvalue* indicates the number of frames on the available frame queue when stealing ends. You can monitor actual conditions on the RMF™ Paging Activity Report (RMF Monitor 1) or equivalent performance monitoring product and adjust accordingly.

  – MCCAECTH

MCCAECTH specifies the low and the OK threshold values for expanded storage. The *lowvalue* indicates the number of frames on the available frame queue when real storage manager (RSM) frame stealing begins. The *okvalue* indicates the number of frames on the available frame queue when stealing ends. You can monitor actual conditions on the RMF Paging Activity Report (RMF Monitor 1) or equivalent performance monitoring product and adjust accordingly.

Note: This parameter is ignored in 64-bit mode.
In 31-bit mode, the defaults are sufficient. For these two parameters, the defaults are MCCAFCTH=(50,100), and MCCAECTH=(150,300). The OK point for available frames in a 31-bit mode implementation is 400 frames, 100 from central storage and 300 from expanded storage.

For 64-bit mode, after installing APARs OW55902 and OW55729, the default values for MCCAFCTH are (400,600). These are IBM's minimum suggested settings. It is suggested that you allow MCCAFCTH to default to (400,600). Higher values are acceptable.

User override: Yes

Reference: See *z/OS MVS Initialization and Tuning Reference*, SA22-7592 for information about the MCCAFCTH and MCCAECTH IEAOPT*xx* parameters, and *z/OS RMF Report Analysis*, SC33-7991 for information about using the Paging Activity report. You should also be familiar with the whitepaper *z/OS Performance: Managing Processor Storage in a 64-bit environment*,WP100269.

- **Couple data set separation**

  Check name: CDS_Dataset_Separation

  Best practice: There are three facets to this check:

  – The primary sysplex, CFRM, and LOGR couple data sets should not reside on the same volume due to the amount of I/O activity for each of these data sets.

  – For all couple data sets, the primary couple data sets should reside on a separate volume from the alternate couple data set.

  – Each primary couple data set has an active alternate couple data set.

  User override: No

  Reference: The publication, *Parallel Sysplex Availability Checklist*, provides detailed recommendations and characteristics about the placement of primary and alternate couple data sets. See also the section, "Planning for the couple data sets," in *z/OS MVS Setting Up a Sysplex*, SA22-7625.

- **Coupling facility structure attributes and location**

  Check name: CouplingFacility_Structure

  Best practice: This check also displays each of the defined coupling facilities, their status, and the relationship between the coupling facilities and structures. The check compares placement based on the preference list, specified in the CFRM policy, which is used to designate the location of coupling facility structures for performance and capacity considerations.

This check shows current status and attributes of each coupling facility. For example, it shows whether a coupling facility is volatile or nonvolatile. Determine if the current status differs from your expectations or requirements.

User override: No

Reference: Refer to the following sections in *z/OS MVS Setting Up a Sysplex*, SA22-7625: ″Understanding preference and exclusion lists″ for information about specification of preferences; ″Using the POPULATECF Function to Rebuild Coupling Facility Structures″ if you want to rebuild any of the coupling facility structures in their preferred coupling facility.

- **Create report for coupling facilities, structures, and systems**

  Check name: Sys_CF_STR_Report

  Best practice: This check produces a report displaying systems, coupling facilities, structures and status of these resources.

  User override: No

  Reference: See the topic about Sysplex policies in *Parallel Sysplex Availability Checklist* for recommendations about structure placement, preferences, and characteristics of structures.

- **Cross system coupling facility (XCF) signalling**

  Check name: XCF_Signalling

  Best practice: This check verifies the following:

  1. all transport classes should be set up to service the pseudo-group name 'UNDESIG'. This ensures that any XCF message can use each transport class.
  2. all defined Transport Classes are assigned to at least one pathout (outbound path).
  3. most pathouts have a transport class defined with a "small" classlength, and at least one other transport class is defined with a higher "large" classlength.
  4. multiple pathout/pathin pairs are in the operational state for each system in the sysplex that is connected to the current system.
  5. a MAXMSG value of a minimum size is defined for each transport class.
  6. each inbound signal path can support a minimum number of messages from the sending system.

  These actions avoid a single point of failure. Exception conditions flagged by this check can reflect a hardware or configuration problem.

  User override: Yes

  Reference: See the topic about Sysplex policies in *Parallel Sysplex Availability Checklist*for recommendations about structure placement, preferences, and characteristics of structures.

- **Cross system coupling facility (XCF) structure location**

Check name: XCF_Signalling_Structures_in_CF

Best practice: If multiple XCF signaling structures are in use, then all of them should not reside on the same coupling facility. There should be at least two online, operational links to each coupling facility. Also, there should not be fewer operational links (CHPID) than there are active links. These actions avoid a single point of failure. Conditions flagged by this check can reflect a hardware problem.

User override: No

Reference: See the topic about Sysplex policies in *Parallel Sysplex Availability Checklist*for recommendations about structure placement, preferences, and characteristics of structures.

- **Sysplex console checks**

  The following group of checks is performed:

  – Consoles are assigned names.

    Check name: Console_Names

    Best practice: IBM suggests that MCS, SNA_MCS, and subsystem consoles are assigned names; this reduces the number of console IDs to help address the limit of 99 consoles per sysplex. Console names are specified within the CONSOL*xx* parmlib entry, using the NAME parameter. The assignment of names to consoles is also required to use alternate groups for consoles.

    User override: No

  – Alternate groups are defined for consoles.

    Check name: Alternate_Console_groups

    Best practice: IBM suggests that you define alternate groups for consoles (using the ALTGRP parameter of the CONSOL*xx* parmlib member). This increases availability if there is a console failure. In such cases, MVS will attempt to switch to another console. Specifying alternate groups (ALTGRP) is preferable to the use of alternate consoles (ALTCONS).

    User override: No

  – Consoles on each system have a console with master authority that has been defined with command association.

    Check name: Console_master

    Best practice: IBM suggests there is a console defined with both MASTER authority and command association for each system in the sysplex.

    For the Console_master check to be successful, each system in the sysplex requires a console. If you did not configure your sysplex so that each system has a console, then you should consider disabling this check using the NOCALL parameter. IBM requests feedback on the value of this check.

    User override: No

  – Master console is active

    Check name: Sysplex_master

Best practice: IBM suggests that the Sysplex Master Console is active within the sysplex.

User override: No

- Console message scope and routing codes

  Check name: Console_MSCOPE_and_Routcodes

  Best practice: Due to the potentially high volume of messages that could be received by a console, IBM suggests that consoles limit the messages and routing codes received to that console's functions. This will improve availability by improving performance and preventing buffer shortages. For example, a console that has a multisystem scope, should receive messages specific to that console's function. Conversely, consoles that are configured ROUTCODE(ALL) should limit the scope of messages received to a single system.

  User override: No

- Use of Action message retention facility (AMRF) and retention of eventual action messages

  Check name: AMRF_and_MPF_consistent

  Best practice: IBM performs this check only if you are using AMRF. The messages can be retrieved at a later time (using the DISPLAY R command). Also, eventual action messages should not be retained to keep the message from becoming too long.

  User override: No

- No console is receiving route code 11 messages

  Check name: Console_routcode_11

  Best practice: Operator consoles do not need to receive routing code 11, which are system programmer messages. This keeps unnecessary messages from being delivered to a console. Route code 11 messages can be retrieved using the DISPLAY R,CE command.

  User override: No

Reference: See *z/OS MVS Planning: Operations*, SA22-7601; the Consoles topic in *Parallel Sysplex Availability Checklist*; and *Parallel Sysplex Managing Software for Availability*, SG24-5451.

- **Extended master console (EMCS) checks**

  To extend the number of consoles on MVS systems, or to allow applications and programs to access MVS messages and send commands, an installation can use extended MCS consoles. The use of these consoles can help alleviate the constraint of the 99 MCS console limit. Moving to an extended MCS console base from a subsystem-allocatable console base will allow for easier expansion in a sysplex.

  Once an EMCS console is defined and activated, it lives for the life of the sysplex-whether it remains active or not. After the number of EMCS consoles (including inactive consoles) becomes very large, console initialization during IPL can be elongeated by minutes. This may occur due to an error in NetView setup or if a CLIST does not reuse EMCS console names. This results in an on-going increase in the number of EMCS consoles defined.

  - Extended master console messages scope and routing codes

Check name: EMCS_Mscope_and_Routcode

Best practice: Due to the potentially high volume of messages that could be received by a console, IBM suggests that EMCS consoles limit the messages and routing codes received to that console's functions. This will improve availability by improving performance and preventing buffer shortages. For example, if an EMCS console is receiving messages from multiple systems, limit the number of route codes assigned to this console. You should not specify ROUTCODE(ALL). Conversely, if an EMCS console is intended to receive all route codes, the scope should be limited to a single system.

User override: No

– Extended consoles with master authority should not be allowed to receive hardcopy messages or to be backup devices for hardcopy medium.

Check name: EMCS_hardcopy

Best practice: An EMCS console should not be defined to receive hardcopy messages if the message scope (MSCOPE) is greater than 1.

User override: No

– Number of EMCS consoles is within recommended range

Check name: Number_EMCS_consoles

Best practice: If the combined total of active and inactive EMCS consoles is excessive, performance can be impacted.

User override: Yes

Reference: See: *z/OS MVS Planning: Operations*, SA22-7601; the Consoles topic in *Parallel Sysplex Availability Checklist*; and *Parallel Sysplex Managing Software for Availability*, SG24-5451.

- **MVS system console checks**

The following group of checks is performed:

– System console is defined to have a local message scope

Check name: SYSCONS_MSCOPE

Best practice: MVS system consoles should be defined to have a local message scope. This reduces the amount of message traffic and improves performance and availability. This is of particular importance when the MVS system console is used during recovery actions.

User override: No

– System console is defined to have a limited number of routing codes

Check name: SYSCONS_ROUTCODES

Best practice: MVS system consoles should be defined to have a limited set of routing codes. This reduces the amount of message traffic and improves performance and availability. The console should be defined with either ROUTCODE(1,2,10) or ROUTCODE(NONE). This is of particular importance when the MVS system console is used during recovery actions.

User override: No

– System consoles are not running in problem determination mode

Check name: SYSCONS_PD_MODE

Best practice: System consoles should not be running in problem determination mode during normal operations. Problem determination mode degrades performance.

User override: No

– Active system console is defined with MASTER authority

Check name: SYSCONS_MASTER

Best practice: The active MCS system console should be defined to have MASTER authority. This is of particular importance when the MVS system console is used as a backup to the sysplex master console.

User override: No

Reference: See: *z/OS MVS Planning: Operations*, SA22-7601; Consoles topic in *Parallel Sysplex Availability Checklist*; and *Parallel Sysplex Managing Software for Availability*, SG24-5451.

- **Real storage settings**

  Check name: Real_Storage_Availability

  Best practice: IBM suggests that both the real and reconfigurable storage parameters should be set to 0. However, this would not be valid if you need to reconfigure storage or to run V=R jobs. The IEASYS*xx* parmlib member should specify the REAL parameter as REAL=0. This will improve performance.

  User override: No

  Reference: See *z/OS MVS Initialization and Tuning Reference*, SA22-7592.

- **Reconfigurable storage settings**

  Check name: RSU_Storage_Availability

  Best practice: IBM suggests that both the real and reconfigurable storage parameters should be set to 0. RSU reflects the amount of central storage to be made available for storage reconfiguration. The IEASYS*xx* parmlib member should specify the RSU parameter as RSU=0.

  User override: No

  Reference: See *z/OS MVS Initialization and Tuning Reference*, SA22-7592.

- **XCF cleanup value**

  Check name: XCF_Cleanup_Value

  Best practice: You should specify a value of 15 for the CLEANUP parameter in the COUPLE*xx* parmlib member. Cleanup specifies how many seconds the system waits for before notifying members that this system is terminating, and loading a nonrestartable wait state. This is the amount of time that members of the sysplex have to perform cleanup processing.

  User override: Yes

  Reference: See *z/OS MVS Initialization and Tuning Reference*, SA22-7592.

- **Sysplex failure detection interval**

  Check name: XCF_Failure_Detection_Interval

Best practice: The CLEANUP INTERVAL parameter in the COUPLE *xx* parmlib member must be coordinated with the spin recovery actions (SPINRCVY) statement in the EXSPAT*xx* parmlib member. The HealthChecker checks that the CLEANUP INTERVAL value conforms to the IBM formula (2*SPINRCVT+5).

The spintime should be defined as 10 seconds for a system in either basic mode or LPAR mode with dedicated CPs. The spintime should be defined as 40 seconds for LPAR mode when the CPs are shared.

User override: Yes

Reference: For information about the EXSPAT*xx* parmlib member, refer to *z/OS MVS Initialization and Tuning Reference*, SA22-7592 and to *z/OS MVS Setting Up a Sysplex*, SA22-7625.

- **Sysplex failure management (SFM) is active**

  Check name: XCF_SYSPLEX_Failure_Management

  Best practice: IBM suggests that you use sysplex failure management (SFM) to define actions to be performed in the event of:

  – Signaling connectivity failures in the sysplex

  – System failures, indicated by a status update missing condition

  – The need to reconfigure systems in a PR/SM™ environment.

  User override: Yes

  Reference: See *z/OS MVS Setting Up a Sysplex*, SA22-7625 for information about defining SFM policies and the *Parallel Sysplex Availability Checklist*.

- **SVC- dump is using dynamically allocated data sets**

  Check name: SDUMP_Availability

  Best practice: IBM suggests that you use dynamic allocation for your dump data sets to ensure that complete diagnostic data is captured at the first occurrence. If your dump data sets are not dynamically allocated and become full, you can lose important diagnostic information.

  User override: No

  Reference: See *z/OS MVS Diagnosis: Tools and Service Aids*, GA22-7589 for information about allocating stand-alone dump data sets.

- **Global Resource Serialization (GRS) STAR configuration**

  Check name: GRS_Mode

  Best practice: A STAR configuration is recommended due to the advantages that it provides with regard to availability, real storage consumption, processing capacity, and response time.

  User override: Yes

  Reference: See *z/OS MVS Planning: Global Resource Serialization*, SA22-7600.

Appendix F. Response messages, error strings, condition codes

Response messages (AOFA0000 — AOFA0099)

Automation returns the following messages to indicate command-invocation, parameter-list, or parameter-resolution problems.

AOFA0000

Explanation: This response message is returned as an indicator for command-invocation, parameter-list, or parameter-resolution problems. If the AOFA0000 response message is returned from the INGHWCMD command list, its data portion is an error string (see Table 18). If the AOFA0000 response message is returned from the INGHWCOM communication task command processor, its data portion contains a condition code from 001 through 033 (see “Hardware Communication Task condition codes "00B00*xxx*"” on page 349).

Table 18. AOFA0000 response message error strings

Error type	Error strings
Environment error	Required_System_Automation _Environment_is_not_complete
Parm error	Proc_or_Sys_name_and_HW_function_name_is_required p_session_type_not_SYNC/ASYNC/blank Timeout_range_*tttt*_already_defined. Timeout_*tt*_out_of_range_1-59. Timeout_specification_range_tt_is_not_valid. Timeout_specification_*tt*_is_not_valid. Timeout_specification_*tttt*_ends_invalid. TRACE_option_must_be_ON_or_OFF *hwcmd*_with_FORCE_operand_is_not_valid. *hwcmd*_is_not_a_supported_HW_function. *p*_must_be_a decimal_integer_value_or_ALL *p*_EXTERNAL_CPU_definition_error *CN*_Activation_profile_name_not_alphanumeric *CN*_Profile_name_is_a_positional_parm *p*_does_not_support_target_wildcard. *p*_Parm_is_in_wrong_position. *lparm*_load_parm_length_must_be_8. *devnum*_device_address_not_hexadecimal. *devnum*_mandatory_load_address_invalid. *lval*_Load_value_definition_error. *lval*_Load_value_is_a_positional_parm. *pn*_load_profile_name_not_alphanumeric. *pn*_Load_profile_definition_error. *pn*_Profile_name_is_a_positional_parm. *spc*_P_and_LV_specs_are_mutually_exclusive. *name*_invalid_chars_in_proc_or_sys_name. *name*_name_longer_than_8_characters. *parm*_Parm_is_unknown_or_in_wrong_position. *evt*_is_an_invalid_event_type *evt*_defined_more_than_once ALL_must_be_the_1st_or_unique_event_parm Event_specification_is_required

Table 18. AOFA0000 response message error strings (continued)

Error type	Error strings
Resolve error	`*sysname*_for_CFs_LOAD/SYSRESET_are_not_supported.` *sysname*_type_specification_missing_or_invalid. *sysname*_null_string_BCP_command_error *pname*_has_invalid_CPC_address_format. *pname*_has_no_IP_address_defined. *pname*__has_no_processor_address_defined. *name*_name_is_not_defined. *pname*_name_not_valid_for_CPC_command. *pname*_has_no_AUTHTKN_defined. No_HWOPER_task_defined.
Check Task	*hwtask*_reached_QueueLimit_qlim *hwtask*_task_msqqeue_data_is_invalid *hwtask*_task_is_not_available *hwtask*_task_module_INGHWCOM_not_running
Hardware Interface	BCP_internal_interface_is_disabled_or_not_active BCP_internal_interface_status_cannot_be_determined
Authorization error	*hwcmd*_has_undefined_access_level *acclevel*_to_*resname*_not_allowed_for_*user* *BadRC*_during_access_chk_for_*resname*

Examples: 1. The INGHWCMD command failed returning an AOFA0000 error string:

```
INGHWCMD MYSYS GETISTAT

AOFA0000 Resolve error:
"MYSYS"_is_not_a_predefined_system_or_CF-name
```

2. The INGHWCMD command failed. Message AOFA0000 was returned by INGHWCOM command processor. The condition code 00B00003 indicates that an unknown communication interface name was passed from INGHWCMD to INGHWCOM.

```
INGHWCMD SC50 GETISTAT

AOFA0000 GETISTAT STATUS(REJECTED) CONDITION(00B00003) SENSE() CPCSNAME()
TSTIME(020111073708)
```

AOFA0001

Explanation: This response message is returned from a request of the following hardware command functions: ACTIVATE, DEACTIVATE, LOAD, RESTART, SYSRESET, START, STOP, CBU, EXTERNAL

Examples: 1. The hardware function STOP was successfully performed for system KEY7:

```
INGHWCMD KEY7 STOP

AOFA0001 STOP KEY7 STATUS(SUCCESS)
CPCSNAME(DEIBMD1.X7E1FA0A)TSTIME(020111135810)
```

2. The hardware function SYSRESET was rejected by INGHWCOM. The condition code 00B00056 indicates that system KEY6 is still operational and cannot be disrupted. In order to perform a disruptive hardware operation, the FORCE option must be specified:

```
INGHWCMD KEY6 SYSRESET

AOFA0001 SYSRESET KEY6
CONDITION(00B00056)SENSE()CPCSNAME(DEIBMD1.X7E1FA0A)TSTIME(020111142827)
```

AOFA0002

Explanation: This message is the response to an INITCOM request. INITCOM establishes a session between INGHWCOM and the Processor Support Element of the addressed hardware.

Example: The session between the INGHWCOM and the processor Support Element of the CPC DEIBMD1.X7E1FA0A, configured with the hardware name ″YORAMA,″ is established successfully:

```
INGHWCMD YORAMA INITCOM

AOFA0002 INITCOM YORAMA STATUS(SUCCESS)
CPCSNAME(DEIBMD1.X7E1FA0A)TSTIME(020111143851)
```

AOFA0003

Explanation: This message is the response to an INITCOM request to an HMC. INITCOM establishes a session between INGHWCOM and the addressed CPC. In case the CPC is defined over an HMC, for each CPC managed by the HMC, one extra line is shown in the report.

Example: The session between INGHWCOM and the HMC where SERVER1 is a member, is established successfully. Implicitly, the sessions to the other CPCs of that HMC are also established:

```
INGHWCMD SERVER1 INITCOM

AOFA0003 INITCOM SERVER1 STATUS(SUCCESS) TSTIME(030117084549)
AOFA0003 INITCOM CPCSNAME(DEIBMIPS.IP3T1000)
AOFA0003 INITCOM CPCSNAME(AUIBMQXP.QXPTHES1)
AOFA0003 INITCOM CPCSNAME(AUIBMQXP.QXPTHES9)
AOFA0003 INITCOM CPCSNAME(DEIBMD1.X7E1FA0A)
AOFA0003 INITCOM CPCSNAME(DEIBMD1.X7F1E30A)
AOFA0003 INITCOM CPCSNAME(DEIBMD1.X7F1F20A)
AOFA0003 INITCOM REPORT COMPLETE
```

AOFA0004

Explanation: This report is the response to a TERMCOM request. TERMCOM ends a session between INGHWCOM and the Processor Support Element of the addressed hardware.

Examples: **1.** The session between the INGHWCOM and the Processor Support Element of the CPC USIBMSC.SCZP107 configured with the hardware name ″P701″ is terminated successfully:

```
INGHWCMD P701 TERMCOM

AOFA0004 TERMCOM P701 STATUS(SUCCESS)
CPCSNAME(USIBMSC.SCZP701)TSTIME(020111090930)
```

2. The session termination between the INGHWCOM and the Processor Support Element of the CPC USIBMSC.SCZP701 configured with the hardware name ″P701″ was rejected. Condition code 00B00033 indicates that no session existed to terminate:

```
INGHWCMD P701 TERMCOM

AOFA0004 TERMCOM P701 STATUS(REJECTED) CONDITION(00B00033) SENSE()
         CPCSNAME(USIBMSC.SCZP701) TSTIME(020111091447)
```

AOFA0005

Explanation: This report is the response to a TERMCOM request for a CPC, which is defined over an HMC connection. In this case, TERMCOM terminates the session between INGHWCOM and the HMC.

Example: The session between INGHWCOM and the HMC where SERVER1 is defined, is terminated successfully. Implicitly the sessions the other CPCs of that HMC are also terminated.

```
INGHWCMD SERVER1 TERMCOM

AOFA0005 TERMCOM SERVER1 STATUS(SUCCESS)
         TSTIME(030117085107)
AOFA0005 TERMCOM CPCSNAME(DEIBMIPS.IP3T1000)
```

```
   AOFA0005 TERMCOM CPCSNAME(AUIBMQXP.QXPTHES1)
   AOFA0005 TERMCOM CPCSNAME(AUIBMQXP.QXPTHES9)
   AOFA0005 TERMCOM CPCSNAME(DEIBMD1.X7E1FA0A)
   AOFA0005 TERMCOM CPCSNAME(DEIBMD1.X7F1E30A)
   AOFA0005 TERMCOM CPCSNAME(DEIBMD1.X7F1F20A)
   AOFA0005 TERMCOM REPORT COMPLETE
```

AOFA0016

Explanation: This report is the response to a CPCDATA request. It returns a report consisting of multiple AOFA0016 messages. The CPCDATA request combines the GETSINFO request for a CPC with the list of GETIINFO request, one for each image of the CPC.

Examples: **1.** The AOFA0016 report message consists of three line record types. The first line type is always the CPC report and the last line is always the report completion type. In between, 1-*n* IMAGE report lines may be displayed, depending on the number of images that are defined for the CPC.

```
   AOFA0016 CPCDATA FREEWAY STATUS(OPERATING)
              PDATA(TYPE(2064),MODEL(107),S/N(000020051528))
              APROF(DEFAULT) MODE(LPAR) CPCSNAME(DEIBMD1.X7F1E30A)
                    TIME(020701083608)
   AOFA0016 CPCDATA CPCINAME(CF1) STATUS(OPERATING) INUMBER(09) IDATA() MODE(CF)
   AOFA0016 CPCDATA CPCINAME(KEY3)STATUS(OPERATING)
              INUMBER(0A)
              IDATA(OSNAME(KEY3),OSTYPE(MVS),OSLEVEL(V1R2),SYSPLEX(KEY1PLEX))
                    MODE(ESA390)
   AOFA0016 CPCDATA CPCINAME(VMA) STATUS(OPERATING) INUMBER(0B)
             IDATA(OSNAME(BOEVMA),OSTYPE(VM)) MODE(ESA390)
   AOFA0016 CPCDATA CPCINAME(DER1) STATUS(OPERATING) INUMBER(0C) IDATA() MODE(LINUXONLY)
   AOFA0016 CPCDATA CPCINAME(DER2) STATUS(NOT_OPERATING) INUMBER(0D) IDATA() MODE(LINUXONLY)
   AOFA0016 CPCDATA REPORT COMPLETE
```

The STATUS field of line one, the CPC status, can have the following values:

```
OPERATING
NOT_OPERATING
NO_POWER
STATUS_CHECK
EXCEPTIONS
POWERSAVE
SERVICE
LINKNOTACTIVE
SERVICE_REQ
UNKNOWN
```

The PDATA field of line one contains the type, model, and serial number of the CPC.

The APROF field of line one contains the last activation profile name used to activate the CPC.

The MODE field of line one, the CPC mode, can have the following values:

```
ESA390
S370
FM
FMAE
HM
HMEA
LPAR
ESA390TPF
CF
FMEX
HMAS
LINUXONLY
```

For each identified CPC image, an AOFA0016 report line is generated.

The CPCINAME field of an image report line contains the image name of an identified image.

The INUMBER field contains the two hex digit partition number. For processor hardware supporting a single channel subsystem, the first digit is always zero. The second digit contains a partition number from 1-F. For processor hardware supporting multiple channel subsystems, the first hex digit contains the channel subsystem number, starting with zero and the second digit contains the partition number 1-F.

The IDATA field contains a collection of the available information supplied by the image BCP. This information can be: `OSNAME`, `OSTYPE`, or `OSLEVEL`; for BCPs of type MVS it can be `SYSPLEX`. Note that one or more IDATA fields may not be available in the AOFA0016, AOFA0017 response reports. This is because not all BCPs may supply the complete field set. If the OSLEVEL field is not shown in the response report, the BCP did not provide this information to the hardware.

The STATUS field of an image report line contains the same status values as supplied with the GETISTAT report message AOFA0017.

2. The processor defined as YORAMA was initialized in ESA390 mode and has a NOT_OPERATING status. The last ACTIVATE was performed using profile KEY6BASIC. Due to its non operational status, no BCP information is available in the IDATA field.

```
AOFA0016 CPCDATA YORAMA STATUS(NOT_OPERATING)
                 PDATA(TYPE(9672),MODEL(RX4),S/N(000510064523)) MODE(ESA390)
                 APROF(KEY6BASIC) CPCSNAME(DEIBMD1.X7E1FA0A) TSTIME(020703094909)
AOFA0016 CPCDATA CPCINAME(X7E1FA0A:Image) IDATA()
AOFA0016 CPCDATA REPORT COMPLETE
```

AOFA0017

Explanation: This report is the response to the following requests: GETSSTAT, GETSDGR, GETSINF, GETISTAT,GETIINFO, and CBU STATUS request.

GETISTAT queries the status of an image object

GETIINFO queries the status of an image object and lists the available image information (OSname,OStype,OSlevel, SysplexName)

GETSSTAT queries the status of a CPC object

GETSDGR queries the degraded reason indicator of a CPC supporting the DEGRADED status

GETSINFO queries the status of a CPC object and lists the available CPC information (machine type and model, CPC serial number, last used activation profile name)

CBU STATUS The status function of the CBU command returns the determined status of the optional capacity backup processor HW feature

On successfull completion, the status field of msg AOFA0017 may have one of the following values:

```
GETIINFO
GETISTAT         CBU STATUS       GETSDGR
-------------    -------------    ----------------
OPERATING        NOT_INSTALLED    NOT_DEGRADED
NOT_OPERATING    NOT_ACTIVATED    MEM_REDUCED
NOT_ACTIVATED    NOT_ENABLED      MEM_BUS_FAILURE
STATUS_CHECK     UNAVAILABLE      NODE_NOT_RUNNING
EXCEPTIONS       AVAILABLE        RING_OPEN
POWERSAVE        ACTIVATED        CBU_EXPIRATION
                                  MRU_FAILURE
                                  TEMPERATURE_PROBLEM
                                  IML_WAS_IN_DEGRADED_MODE
```

Examples: **1.** The own system ('*'), which runs on LPAR A3 of CPC USIBMSC.SCZP801, has a status of OPERATING and its system name is SC50.

```
   INGHWCMD * GETISTAT

   AOFA0017 GETISTAT SC50 STATUS(OPERATING)CPCINAME(A3)CPCSNAME(USIBMSC.SCZP801)
            TSTIME(020111095940)
```

2. On processor YORAMA, the logical partition KEY7 has a status of of OPERATING:

```
INGHWCMD YORAMA.KEY7 GETISTAT

AOFA0017 GETISTAT YORAMA.KEY7 STATUS(OPERATING) CPCINAME(KEY7)
         CPCSNAME(DEIBMD1.X7E1FA0A) TSTIME(020204130403)
```

3. On processor FREEWAY, Capacity Backup Upgrade is installed and enabled (AVAILABLE). It was ACTIVATED on the 18th of April 2002 and will expire on the 9th of September 2002. There are four test-activations left for processing.

```
INGHWCMD FREEWAY CBU STATUS

AOFA0017 CBU FREEWAY STATUS(AVAILABLE,ACTIVATED)
         ACTIVATION(18/04/02) EXPIRATION(09/09/02) TESTSLEFT(4)
         CPCSNAME(DEIBMD1.X7E1FA0A)
         TSTIME(020204130403)
```

4. The system defined as KEY6 to msys for Operations, which runs on logical partition KEY6 (cpciname), whose partition number cannot be determined (inumber field is empty), is an MVS OS type with an OS defined name KEY6. It runs as a memebr of the sysplex KEY6PLEX. The Lpar runs in ESA mode and is running on CPC DEIBMD1.X7E1FA0A.

```
INGHWCMD KEY6 GETIINFO

AOFA0017 GETIINFO KEY6 STATUS(OPERATING) CPCINAME(KEY6)
         INUMBER(0A) IDATA(OSNAME(KEY6),OSTYPE(MVS),SYSPLEX(KEY6PLEX))
         MODE(ESA) CPCSNAME(DEIBMD1.X7E1FA0A)
         TSTIME(020923110403)
```

AOFA0018

Explanation: This report is returned in response to a GETCLUSTER command.

Examples: **1.** From the own system (’*’), the CPC addresses list in PDATA are in your scope of control. With a BCP Internal Interface connections, this list is determined internally from the local SE by contacting the HMC in your processor LAN that has the ″Change Management″ function enabled. The content of the Defined CPCs group of this HMC represents the CPCs that you can contact through this BCP Internal Interface session. Each scope list is terminated with a ″report complete″ message. The PDATA field of the AOFA0018 message contains CPC-related information. The first PDATA entry is always the fully qualified address of the CPC (cpcsname). Other PDATA information may be added in the future, separated by a comma.

```
INGHWCMD * GETCLUSTER

AOFA0018 GETCLUSTER SC50 STATUS(SUCCESS)
CPCSNAME(USIBMSC.SCZP801)TSTIME(020112054842)
AOFA0018 GETCLUSTER PDATA(USIBMSC.SCZP801)
AOFA0018 GETCLUSTER PDATA(USIBMSC.SCZP701)
AOFA0018 GETCLUSTER PDATA(USIBMSC.SCZP702)
AOFA0018 GETCLUSTER PDATA(USIBMSC.SCZP601)
AOFA0018 GETCLUSTER REPORT COMPLETE
```

2. This GETCLUSTER request failed with a condition code of 0B100224 representing a BCP Internal Interface transport timeout condition:

```
INGHWCMD * GETCLUSTER

AOFA0018 GETCLUSTER SC50 STATUS(FAILED) CONDITION(0B100224)
                    SENSE(00000000 0000 00000000)
                    CPCSNAME(USIBMSC.SCZP801)TSTIME(020111085916)
```

AOFA0019

Explanation: This report is returned response to a FILTER LIST command.

Examples: **1.** The filter list report shows the filters that are in place for SERVER1 CPC. The defined events in the list (EN) are only forwarded to the specified NetView operators or operator group, if at least one filter is set. If a prefix field PFX was specified with a filter SET command, its text is placed in front of every event message (ISQ900I).

```
INGHWCMD SERVER1 FILTER LIST

AOFA0019 FILTER SERVER1 STATUS(SUCCESS) CPCSNAME(DEIBMD1.X7F1F20A) TSTIME(030128090248)
AOFA0019 FILTER CPC PDATA(EN(ST,HW) OP(TIL) PFX(ISQ900I))
AOFA0019 FILTER CPC PDATA(EN(CC,ST,HW,ALRT) OP(+A00001S) PFX(ISQ900I))
AOFA0019 FILTER CPC PDATA(EN(CC,ST,HW) OP(+O00001S) PFX(ISQ900I))
AOFA0019 FILTER REPORT COMPLETE
```

If a prefix field PFX was specified with a filter SET command, its text is placed in front of every event message. If no prefix is given with a FILTER SET command, the default prefix AOFA0900 is used.

2. The filter list report shows the filters that are in place for image KEY2 on CPC SERVER1.

The first filter set sends all event reports to group +GEOOPER in a event message prefixed with GEO001I. The second filter sends the event messages with the default prefix AOFA0900, because in the preceeding filter set command PFX was not specified.

```
INGHWCMD SERVER1.KEY2 FILTER LIST

AOFA0019 FILTER SERVER1.KEY2 STATUS(SUCCESS) CPCSNAME(DEIBMD1.X7F1F20A) TSTIME(030119070203)
AOFA0019 FILTER INAME(KEY2) PDATA(EN(CC,ST,HW,BCP,ALRT) OP(+GEOOPER) PFX(GEO001I))
AOFA0019 FILTER INAME(KEY2) PDATA(EN(CC,ST,HW,BCP) OP(TIL) PFX(AOFA0900))
AOFA0019 FILTER REPORT COMPLETE
```

AOFA0099

Explanation: This report is returned as the response to a connection status request command.

Examples: **1.** The CPC SERVER1 is connected to this NetView using task AUTHW007. The connection is made either through INTERNAL or SNMP tranport. The BCP Internal Interface uses the internal transport and always has the Support Element (SE) as target. For the SNMP tranport, either SE or HMC can be the target. A TRANSPORT value of A-INTERNAL indicates an asynchronous session, whereas S-INTERNAL indicates a synchronous session. A value of SNMP indicates that a standard SNMP session is established, which is always asynchronous.

```
INGHWCMD SERVER1 STATCOM

AOFA0099 STATCOM SERVER1 STATUS(CONNECTED)
                 TASK(AUTHW007)
                 TRANSPORT(A-INTERNAL)
                 TARGET(SE)
                 CPCSNAME(IBM390PS.P1234567)
                 TSTIME(030104091131)
```

2. The CPC SAFOS is currently not connected to the INGHWCMD hardware interface.

```
INGHWCMD SAFOS STATCOM

AOFA0099 STATCOM SAFOS STATUS(NOT_CONNECTED)
                 CPCSNAME(DEIBMD1.X7F1F20A)
                 TSTIME(030104091145)
```

3. The CPC YORAMA is currently connected to the INGHWCMD hardware interface with a synchronous SNMP session. Synchronous sessions (SNMP or INTERNAL) allow the polling of CPC and image status and the retrieval of object information. HW events, such as status changes or messages from the HW or BCPs, cannot be processed with synchronous sessions.

```
INGHWCMD YORAMA STATCOM

AOFA0099 STATCOM SAFOS STATUS(CONNECTED)
                 TASK(ISQCM001)
                 TRANSPORT(S-SNMP)
                 TARGET(SE)
                 CPCSNAME(DEIBMIP1.IP3T1100)
                 TSTIME(030104091145)
```

Asynchronous response messages (AOFA0100-AOFA0900)

Events from CPCs or CPC images arrive as asynchronous messages in NetView. The following event types are supported by INGHWCMD interface:

- Messages from the Operating System (BCP messages)
- Status Changes of CPC or CPC image objects
- Hardware Messages from the CPC and its associated CPC images
- SNA Alert data from the CPC and its associated CPC images

AOFA0100 NEWSTATUS(*obj_status_number***) OLDSTATUS(***obj_status_number***)**

Explanation: This message indicates a status change event of a CPC or CPC image object. Before you can receive this type of message, you must have successfully initialized an asynchronous session. Secondly, you must have specified a FILTER command with a NetView operator or group name as the receiver for each image or CPC that you target. See "Hardware object status summary" on page 362 for a list of the supported object status numbers and their meaning.

Example: Partition KEY4 of FREEWAY was successfully CP stopped. The status changed from OPERATING (0001) to NOT_OPERATING (0002).

After the LPAR was CP started again, its status went back to OPERATING. Note that Prefix message AOFA0900 was used, because the FILTER command was specified without a PFX. No operating system messages are shown either, because the event notification that was enabled with the FILTER request was only ST, for object status change events.

```
INGHWCMD FREEWAY INITCOM ASYNC
INGHWCMD FREEWAY.KEY4 FILTER SET(ST) OP(TIL)
INGHWCMD FREEWAY.KEY4 STOP

AOFA0001 STOP FREEWAY.KEY4 STATUS(ACCEPTED) CPCINAME(KEY4) CPCSNAME(DEIBMD1.X7F1E30A)
              TSTIME(030208143117)

AOFA0900 FREEWAY.KEY4 SC AOFA0100 NEWSTATUS(0002) OLDSTATUS(0001)

INGHWCMD FREEWAY.KEY4 START

AOFA0001 START FREEWAY.KEY4 STATUS(ACCEPTED) CPCINAME(KEY4) CPCSNAME(DEIBMD1.X7F1E30A)
               TSTIME(030208143134)

AOFA0900 FREEWAY.KEY4 SC AOFA0100 NEWSTATUS(0001) OLDSTATUS(0002)
```

AOFA0200 *CpcName.ImageName ConID Command_completion_response*

Explanation: This response message uses the following variables:

CpcName.ImageName
Specifies the name of the processor as defined in AOFCUST and the image or LPAR name that this asynchronous command completion message originates from.

ConID Always SC for System Console.

Command_completion_response
Message string providing the command name, completion status, and possible condition code information of an HW command.

AOFA0300 *CpcName ConID Alert_information*

Explanation: This response message uses the following variables:

CpcName Specifies the name of the processor as defined in AOFCUST.

ConID Always SC for System Console.

Alert_information Message string providing containing alert details.

AOFA0400 *CpcName ConID HW_message*

Explanation: This response message uses the following variables:

CpcName Specifies the name of the processor as defined in AOFCUST.

ConID Always SC for System Console.

HW_message The message text is identical to the short message text displayed on the SE or HMC, when a CPC object has been marked, for which an HW message is waiting. Note that the message detail information can only be accessed using the HMC/SE GUIs.

AOFA0900 *CpcName.ImageName ConID Event_message_string*

Explanation: This response message uses the following variables:

CpcName.ImageName
Specifies the name of the processor as defined in AOFCUST and the image or LPAR name that this event message originates from.

ConID For BCP message events the id is OC, indicating a message from the operator console. For all other event messages, the id is SC, indicating a message from the system console.

Event_message_string
Specifies the message text from the operating system as shown on the SE or HMC, or the message string specific to the other events type.

AOFA0998 **EVENT HANDLER WARNING: ASYNCHRONOUS SESSION HAS BEEN REESTABLISHED**

Explanation: The HW event handler for this SE/HMC recovered a missing heartbeat condition signaled by the BCP Internal Interface. Some events might have been lost. Applications should query status information of the CPC or image in order to determine the current status. Note that BCP messages that are transported as HW events are not buffered and cannot be recovered if they are lost.

AOFA0999 **EVENT HANDLER ERROR RC(*rc*) , ASYNCHRONOUS SESSION IS TERMINATING.**

Explanation: The HW event handler for this SE or HMC session returned an error return code (*rc*). The asynchronous session terminates. An INITCOM ASNC must be issued to reactivate the session. Refer to “Data exchange services ″0B100*xxx*″” on page 352 if the return code number is less than 100. Refer to “Internal transport services ″0B*x*00*xxx*″” on page 355 for return codes greater than100. Note that only if you set a CPC event filter will you receive this event handler termination report.

Condition codes

This section gives further information about the condition codes for errors associated with the following:

- “Hardware Communication Task condition codes ″00B00*xxx*″”
- “Data exchange services ″0B100*xxx*″” on page 352
- “Command services ″0B200*xxx*″” on page 354
- “Internal transport services ″0B*x*00*xxx*″” on page 355

Hardware Communication Task condition codes ″00B00*xxx*″

Table 19 lists the condition codes for Hardware Communication Task ″00B00*xxx*″.

Table 19. Hardware Communication Task condition codes

Reason Code	Error String	Error Description
001	`ING_invalid_HLL_buffer`	INGHWCOM was invoked, but the NetView HLL buffer found for C/C++ is not valid.

Table 19. Hardware Communication Task condition codes (continued)

Reason Code	Error String	Error Description
002	ING_origuser_invalid	The userid and output correlator passed to INGHWCOM is not valid.
003	ING_interface_invalid	The hardware interface name passed to INGHWCOM is not valid. Allowed interface names are INTERNAL, SNMP, or SNA.
004	ING_interface_missing	No hardware interface name is passed to INGHWCOM.
005	ING_tgt_length-error	Parsing Error: The target object name (processor or image name) has an invalid length. It must be 1 to 8 characters.
006	ING_tgt_missing	Parsing Error: The target object name (processor or image name) is not specified.
007	ING_cpc_length_error	Parsing Error: The CPC address specification netid.nau has an invalid length. It must not exceed 17 characters.
008	ING_cpc_missing	Parsing Error: The CPC address specification, which is a required parameter for the request, is missing.
009	ING_imgname_length_err.	Parsing Error: The image name (Lpar name) parameter has an invalid length.
00A	ING_imgname_missing	Parsing Error: The image name (Lpar name) is a required parameter for the request, but has not been specified.
00B	ING_force_invalid	Parsing Error: The FORCE option is specified in the request but is not supported for the HW function. The following HW functions allow the FORCE option: ACTIVATE, DEACTIVATE, SYSRESET, LOAD
00C	ING_force_missing	Parsing Error: The FORCE option is required for the request, but has not been specified.
00D	ING_auth_missing	Parsing Error: The AUTHENTICATION specification that is required for each request is missing.
00E	ING_timeout_missing	Parsing Error: The required TIMEOUT parameter is missing in the request.
00F	ING_OCFCMD_truncated	Parsing Error: The HW function (OCFCMD) exceeds the maximum allowed length, which is 40 characters.
010	ING_OCFCMD_missing	Parsing Error: No HW function (OCFCMD) was specified in the request.
020	ING_SNMP_noIP_address	Parameter Resolution Error: The SNMP interface was specified for the HW request, but no IP address information is available.
021	ING_OCF_resolve_failed	Parameter Resolution Error: No HW function name to resolve, same as error 010.
022	ING_OCF_not_resolved	Parameter Resolution Error: An invalid HW function name was detected.
023	ING_BCP_null_cmd	Parameter Resolution Error: The BCP command retrieved from NetView Cglobal ING.*xxxx*.CMDTXT was empty.
024	ING_BCP_cmd_acc	Parameter Resolution Error: The NetView Cglobal variable containing the BCP command could not be retrieved.
025	ING_session_type_missing_or_invalid	Parameter Resolution Error: Session type values SYNC or ASYNC missing, or an invalid session type value was passed.
026	ING_active_session_type_mismatch	Parameter Resolution Error: A session was requested having a different session type SYNC/ASYNC than the active session. Reissue the INITCOM request with the correct session type or terminate the active session prior to requesting a new one.

Table 19. Hardware Communication Task condition codes (continued)

Reason Code	Error String	Error Description
030	`ING_nt_alloc_error`	Storage Allocation Error: The Netid base table could not be allocated using CNMNAMS services.
031	`ING_img_alloc_error`	Storage Allocation Error: The storage for a system image could not be allocated using CNMNAMS services.
032	`ING_img_locate_error`	Storage Allocation Error: The previously allocated storage for a system image could not be located using CNMNAMS services.
033	`ING_notinit_error`	Storage Allocation Error: An HW function request was issued for a processor or system image without having allocated storage for that processor or system image. This happens if no INITCOM request was made prior to the first a HW function request.
050	`ING_notinitized_error`	HW Function Error: An HW function request was issued for a processor or system image without having done an INITCOM. Same as error 033.
051	`ING_imgnotfound_error`	HW Function Error: An HW function request was issued for a system image that could not be located as an image belonging to the addressed CPC.
052	`ING_funcunknown_error`	HW Function Error: An unknown HW function name was requested. Same as error 022.
053	`ING_nocpcobject_error`	HW Function Error: GETCLUSTER failed. Cluster list attribute not resolved by the processor support element.
054	`ING_nocluster_error`	HW Function Error: GETCLUSTER failed. The cluster list returned by the processor support element was empty.
055	`ING_nohwstatus_error`	HW Function Error: An HW function which requires the determination of the status of the object prior to execution cannot be processed because the object status cannot be determined. This error is valid only for processors where the FORCE option has to be emulated by INGHWCOM.
056	`ING_disruptive_cmd`	HW Function Error: A disruptive HW function was requested without the FORCE option and the processor/image object is in an operational state. INGHWCOM uses FORCE(NO) (allow no disruptive commands) as default. If you want to allow disruptive commands you must specify the FORCE option in the INGHWCMD request.
057	`ING_noistatus_error`	HW Function Error: CBU failed. A CBU function was requested but the current CBU status cannot be determined.
058	`ING_noiobject_error`	HW Function Error: CBU failed. The processor hardware does not support the CBU installed object attribute.
059	`ING_cbustatus_error`	HW Function Error: CBU failed. A CBU status was returned that does not allow the request.
060	`ING_filter_error`	HW Function Error: A filter SET/UNSET command failed. Either a filter table is full and no new filters can be set, or a specified image name does not exist on the CPC. A maximum of 10 filter entries can be set per image. This error is also shown when doing a SET for an event type of BCP for a CPC object in LPAR mode. For CPC objects, this type of event is only supported when running in a BASIC processor mode.

Table 19. Hardware Communication Task condition codes (continued)

Reason Code	Error String	Error Description
061	`ING_filter_not_async`	HW Function Error: A filter command was entered for a SYNCHRONOUS session of the BCP Internal Interface, where filter commands are not supported. Synchronous sessions over the BCP Internal Interface are established if the SE MCL does not support asynchronous sessions.
065	`ING_session_not_async`	HW Function Error: A REGISTER request was issued for a SYNCHRONOUS BCP Internal Interface session, which is not supported. In order to register for HW events the sessions must be asynchronous. For SNMP, sessions are implicitly asynchronous. The BCP Internal Interface tries to establish a session in asynchronous mode. If this fails, a synchronous session is tried.
070	`ING_hmccpc_tbl_error`	Initialization Error: An internal HMC/CPC/IMG table error occured during INITCOM processing. Enable AOCTRACE and rerun INITCOM to get additional information about the problem.
071	`ING_cpcimg_alloc_error`	Initialization Error: For an image, the dynamic storage allocation request failed during INITCOM processing. Enable AOCTRACE and rerun INITCOM to get additional information about the problem.
072	`ING_cpc_config_mismatch`	Initialization Error: SNMP connections only. For SE connections , the CPC SNA address returned from the SE is different to the CPC SNA address configured for the CPCname used in the INITCOM request. For HMC connections, the configured SNA address of the CPCname of the INITCOM request is not defined on this HMC. For both connection types the communication is terminated.
0A0	`ING_invalid_task`	HW Task Error: The HW communication interface is running on a NetView task that is not the configured task. Module INGHWCOM terminates. Verify that INGRCUST contains a valid autotask name for the HWOPER02 and HWOPER01 keywords (msys for Operations only).
0A2	`ING_config_error`	HW Task Error: The configuration information about the NetView autotask names to be used for the HW communication interface cannot be retrieved. This happens if the interface is called but msys for Operations initialization is not complete. This error also happens if the autotasks are not defined. See error code 0A0 for additional information.

Data exchange services "0B100*xxx*"

Note that this set of condition codes applies to SNMP connections *only*.

Table 20 on page 353 lists the condition codes that are returned if there is an error with the following INGHWCMD functions:

- INITCOM
- TERMCOM
- ACTIVATE
- DEACTIVATE
- SYSRESET
- START
- STOP

- RESTART
- LOAD
- CBU
- EXTERNAL
- GETSSTAT
- GETSINFO
- GETISTAT

The condition code data "*xxx*" prefixed by 0B100 is returned as part of the following response messages, with a status value of REJECTED or FAILED:

- AOFA0001
- AOFA0002
- AOFA0004
- AOFA0017
- AOFA0018

For additional return code information please refer to the chapter "*Data Exchange API Call Return Codes*," in the Appendix of the most current release of *zSeries 900 Application Programming Interface*, SB10-7030.

Table 20. Data Exchange Services condition codes

Condition Code	Error String
001	`HWMCA_DE_NO_SUCH_OBJECT` If this condition code is returned from an INITCOM request,the community name settings in the Support Element or HMC do not allow access from your location. Check the netmask of the SNMP settings of your SE/HMC. For other request types, this condition indicates that an object (CPC,Image, Profile) or an associate attribute, cannot be found for this request in the SNMP MIB data base of the SE/HMC. A possible cause may be that an object has been requested that is not available on this CPC's type or CPC's microcode level.
002	`HWMCA_DE_INVALID_DATA_TYPE`
003	`HWMCA_DE_INVALID_DATA_LENGTH`
004	`HWMCA_DE_INVALID_DATA_PTR`
005	`HWMCA_DE_INVALID_DATA_VALUE`
006	`HWMCA_DE_INVALID_INIT_PTR`
007	`HWMCA_DE_INVALID_ID_PTR`
008	`HWMCA_DE_INVALID_BUF_PTR`
009	`HWMCA_DE_INVALID_BUF_SIZE`
010	`HWMCA_DE_INVALID_DATATYPE_PTR`
011	`HWMCA_DE_INVALID_TARGET`
012	`HWMCA_DE_INVALID_EVENT_MASK`
013	`HWMCA_DE_INVALID_PARAMETER`
014	`HWMCA_DE_READ_ONLY_OBJECT`
015	`HWMCA_DE_SNMP_INIT_ERROR` This is a retryable condition code.
016	`HWMCA_DE_INVALID_OBJECT_ID`
017	`HWMCA_DE_REQUEST_ALLOC_ERROR`

Table 20. Data Exchange Services condition codes (continued)

Condition Code	Error String
018	HWMCA_DE_REQUEST_SEND_ERROR
019	HWMCA_DE_TIMEOUT
020	HWMCA_DE_REQUEST_RECV_ERROR
021	HWMCA_DE_SNMP_ERROR Check that the SNMP API is enabled on the SE or HMC.
022	HWMCA_DE_INVALID_TIMEOUT
028	HWMCA_DE_INVALID_HOST
029	HWMCA_DE_INVALID_COMMUNITY
030	HWMCA_DE_INVALID_QUALIFIER
098	HWMCA_DE_REQUIRES_QUALIFIER
099	HWMCA_DE_TRANSPORT_ERROR Check that the BCP Internal Interface modules are correctly installed. Module or module version error detected in LPA lib or LNKLST.

Command services "0B200*xxx*"

Table 21 lists the condition codes that are returned if there is an error with the following hardware functions:

- ACTIVATE
- DEACTIVATE
- SYSRESET
- START
- STOP
- RESTART
- LOAD
- CBU
- EXTERNAL

The condition code data *"xxx"* prefixed by 0B200 is returned as part of the AOFA0001 response message with a status value of REJECTED or FAILED.

For additional return code information please refer to the chapter *"Data Exchange API Call Return Codes,"* in the Appendix of the most current release of *zSeries 900 Application Programming Interface*, SB10-7030.

Table 21. Command Services condition codes

Condition Code	Error String
000	HWMCA_CMD_STARTED_BUT_COMPLETION_MAY_HAVE_FAILED An HW command was successfully started. For SYNC sessions check the AOFA0001 command completion report for additional sense information about the actual command completion. With ASYNC sessions, the command completion event report has this information, for which applications can register with the FILTER command.
001	HWMCA_CMD_NO_SUCH_OBJECT An HW command has been requested that is not available on this HW type or microcode level.

Table 21. Command Services condition codes (continued)

Condition Code	Error String
002	HWMCA_CMD_INVALID_DATA_TYPE
003	HWMCA_CMD_INVALID_DATA_LENGTH
004	HWMCA_CMD_INVALID_DATA_PTR
005	HWMCA_CMD_INVALID_DATA_VALUE
006	HWMCA_CMD_INVALID_INIT_PTR
007	HWMCA_CMD_INVALID_ID_PTR
010	HWMCA_CMD_INVALID_DATATYPE_PTR
011	HWMCA_CMD_SNMP_ERROR_INCONSISTENT_NAME A command request was received while a prior request was just finished. The prior task is still not complete, but it has released the RESERVE already. This is a temporary condition and the request can be repeated.
013	HWMCA_CMD_INVALID_PARAMETER
017	HWMCA_CMD_REQUEST_ALLOC_ERROR
018	HWMCA_CMD_REQUEST_SEND_ERROR
019	HWMCA_CMD_TIMEOUT
020	HWMCA_CMD_REQUEST_RECV_ERROR
021	HWMCA_CMD_SNMP_ERROR
022	HWMCA_CMD_INVALID_TIMEOUT
023	HWMCA_CMD_INVALID_CMD
024	HWMCA_CMD_OBJECT_BUSY
025	HWMCA_CMD_INVALID_OBJECT
026	HWMCA_CMD_COMMAND_FAILED
027	HWMCA_CMD_INITTERM_OK
028	HWMCA_CMD_CBU_DISRUPTIVE_OK
029	HWMCA_CMD_CBU_PARTIAL_HW
030	HWMCA_CMD_CBU_NO_SPARES
031	HWMCA_CMD_CBU_TEMPORARY
032	HWMCA_CMD_CBU_NOT_ENABLED
033	HWMCA_CMD_CBU_NOT_AUTHORIZED
034	HWMCA_CMD_CBU_FAILED
035	HWMCA_CMD_CBU_ALREADY_ACTIVE
036	HWMCA_CMD_CBU_INPROGRESS
099	HWMCA_TRANSPORT_ERROR Before the HwmcaCmd was invoked, an internally issued data entry Hwmca call terminated with CC 99. Same as HWMCA_DE_TRANSPORT_ERROR condition code 0B100099.

Internal transport services "0B*x*00*xxx*"

Note that this set of condition codes applies to BCP Internal Interface connections *only*.

Table 22 lists the condition codes that are returned if there is an error with the following INGHWCMD functions:

- INITCOM
- TERMCOM
- ACTIVATE
- DEACTIVATE
- SYSRESET
- START
- STOP
- RESTART
- LOAD
- CBU
- EXTERNAL
- GETSSTAT
- GETISTAT

The condition code data "*xxx*" prefixed by 0B100 or 0B200 is returned as part of the following response messages, with a status value of REJECTED or FAILED:

- AOFA0001
- AOFA0002
- AOFA0004
- AOFA0017
- AOFA0018

Table 22. Internal Transport Services condition codes

Reason Code	Error Description
100	A problem was encountered prior to sending the request to the HSAET32 API for processing. This is likely due to a failure to an environmental error. Check if the Support Element is fully operational. A running reboot of the SE may have caused this problem.
101	A request incomplete condition occured.
102	A report list overflow occurred. This return code should not currently be issued for SNMP requests, however is included for OCF query (Query-Read-Cluster) compatibility.
110	The issuer of the request is not (RACF) authorized to the requested function. Note that (like HCD) the HSAET32 services require that RACF or a compatible SAF product be installed and operational. An RACF system message ICH408I is issued with additional information. If no ICH408I message is issued, make sure the class FACILITY is RACLISTed and repeat the request.
111	The control block ID or version of the HSDB passed to HSAET32 services is invalid. For hwmcaapi requests, this indicates that HSAPHCPI is incompatible with the supporting HSAPHARI module.
112	The requested function is invalid or not supported by the current level of HSAET32 services. For hwmcaapi requests, this indicates an incompatibility between HSAPHCPI and the supporting HSAPHARI module.
113	The control block ID of the request list passed to HSAET32 services is invalid or inappropriate for the requested function. For hwmcaapi requests, this indicates a problem in module HSAPHCPI.

Table 22. Internal Transport Services condition codes (continued)

Reason Code	Error Description
114	The request list entry count passed to HSAET32 services is invalid or inappropriate for the requested function. For hwmcaapi requests, this indicates a problem in module HSAPHCPI.
115	The request list entry pointer passed to HSAET32 services is null and therefore invalid. For hwmcaapi requests, this indicates a problem in module HSAPHCPI.
116	Some of the input areas passed to HSAET32 services exist in a storage area that the caller does not have authority to fetch or update.
117	The input parameter list generated by the HSAXHARI (or CBDIHSD) macro does not have the correct version ID or type, or does not point to an HSDB.
118	The control block ID of the Output Report request list passed to HSAET32 services is invalid or inappropriate for the requested function. For hwmcaapi requests, this indicates a problem in module HSAPHCPI.
119	The session token is invalid. This is probably due to a previous failure of the hwmcainitialize request, an hwmcaterminate request being issued for the session, or improper modification of the HWMCA_SCLP_TARGET_INFO structure.
120	The host environment does not support HSAET32 services. HSAET32 services are not currently on VM hosts.
121	An address space resource manager could not be established.
122	A task resource manager could not be established.
123	The HSAET32 associated recovery routine (HSAPHARR) was entered due to an unexpected error processing the request.
124	The CBDMHWA CSECT could not be found in the nucleus.
125	The HSAET32 monitor exit (HSAPHMON) could not be established as the secondary ET32 listner exit for the application.
126	The system date and time could not be obtained to correlate HRE and associated MDS-MU's.
127	A failure occurred attempting to access the HWAX.
129	An attempt to send the MDS_MU requests across the BCP Internal Interface interface failed.
130	Either the EP_OPERATIONS_MGMT vector (9F22) from the event type 30 data was not available or its length was invalid.
131	Either the application name-group for the EP_OPERATIONS_MGMT application (event type 30 data) was not returned in the 9F22 vector or its length was invalid.
132	Either the NetID of the local support element was not returned in the application name-group for the EP_OPERATIONS_MGMT application (event type 30 data) or its length was invalid.
133	Either the NAU of the local support element was not returned in the application name-group for the EP_OPERATIONS_MGMT application (event type 30 data) or its length was invalid.
134	Either the CPC image name vector (9F70) from the event type 30 data was not available or its length was invalid.
135	Error in ET30 ESTAE routine. Module HSAPHSDI was unable to establish an ESTAEX recovery environment.

Table 22. Internal Transport Services condition codes (continued)

Reason Code	Error Description
136	Either the primary OCF name vector (9F81) from the event type 30 data was not available or its length was invalid.
140	BCP Internal Interface Access Error. HSAET32 services have been disabled or were not been started.
148	BCP Internal Interface Session Error. The hardware session with the target CPC has terminated. This is condition is raised due to a missing heartbeat from the ET32 agent code running on the target SE. To recover from this situation the application should perform a TERMCOM followed by an INITCOM in order to re-establish session communication. The occurence of this RC indicates that possibly events have been lost.
150	Error in ET32 ESTAE routine.
151	HSAPHSPI identified a parameter that is not contextually valid.
152	HSAPHSPI identified a missing parameter that is contextually required.
153	HSAPHSPI identified a parameter value that is syntactically incorrect.
160	IEAMSCHD_Error. An error has occured attempting to schedule HSAPHDSC for execution. An IEAMSCHD retrun code is also provided to further explain the cause of the error.
161	CSS_Error. An error condition was raised using callable supervisor services facilities. The IEAV*xxxx* return code is also provided to further explain the cause of the error.
162	CPSS_Error. An error condition was raised using cellpool services facilities. The CSRP*xxx* return code is also provided to further explain the cause of the error.
163	HSAPHDSC_Error. An error condition was raised using callable supervisor services facilities. The IEAV*xxxx* return code is also provided to help identify the cause of the error.
164	CTRACE_Error. An error condition was encountered by module HSAPHDSC, and is further identified by the return code also provided.
165	An error condition was encountered while attempting define (or delete) the application to Component Trace. The CTRACE return code is also returned to help identify the cause of the error.
166	HSAPHDSC Error. The Access List Service macro ALESERV returned an error condition. Report the available sense data of the AOFA0001 response when contacting IBM support.
167	HSAPHDSC Error. The TCB Token Service macro TCBTOKEN returned an error condition. Report the available sense data of the AOFA0001 response when contacting IBM support.
168	HSAPHDSC/HSAPHMNX Error. The Data Space Service macro DSPSERV returned an error condition. Report the available sense data of the AOFA0001 response when contacting IBM support.
169	HSAPHARI Error. The Component Trace Service returned an error condition. Report the available sense data of the AOFA0001 response when contacting IBM support.
170	HSAPHARI/HSAPHMNX Error. The Callable CellPool Service CSRPEXP returned an error condition. Report the available sense data of the AOFA0001 response when contacting IBM support.

Table 22. Internal Transport Services condition codes (continued)

Reason Code	Error Description
171	HSAPHARI Error. The Name Token Service IEANTCR returned an error condition. Report the available sense data of the AOFA0001 response when contacting IBM support.
204	The request was accepted by the local support element and will be processed asynchronously. No further reason code is provided. (This function is not currently used by the hwmcaapi implementation).
208	Execution of request was failed by the target support element. This indication is normally accompanied by a condition report that is returned as the error reason, and may also be accompanied by sense data further identifying the cause of the failure.
212	The request was rejected by the local support element. This indication is normally accompanied by a condition report that is returned as the error reason.
216	An MDS-MU error message was received from the target support element. The condition report code is returned as the error reason.
220	HSAPHMNX detected a structural error while processing the incoming report from the target Support Element. If the SENSE field contains data, check if the first byte has a value of X'27'. In this case, a request length difference was detected between the data coming from the OS and what was returned from the target Support Element. Note that in this case a LOGREC software symptom record is written containing additional data. (HSAET32)
224	No response was received within the time interval designated for the request. No further reason code provided.
228	An error was detected in a request list entry. An internal reason code is generated to identify the field in error. (This code is not used for hwmcaapi requests).
232	A routing error has occurred while forwarding the requests for processing. This indication is normally accompanied by a condition report that is returned as the error reason.

Appendix G. Sense codes, hardware object status summary

Sense codes

Note that for BCP Internal Interface connections the sense codes are copied from the request response report information into the AOFA*xxxx* messages. For sense data detail information please refer to chapter *"Appendix B. HWMCA_EVENT_COMMAND_RESPONSE Return Codes"* in the most current release of *zSeries 900 Application Programming Interface*, SB10-7030.

For online reference, please use NetView's SENSE command followed by the 8-character sense code parameter from the returned AOFA*xxxx* report string.

0806000A RESOURCE UNKNOWN

Explanation: The profile name (CNAME) specified in a operations command is not recognized by the receiving node.

System Programmer Response: Correct the configuration identifier and resend the request.

08090000 Mode inconsistency: The requested function cannot be performed in the present state of the receiver.

Explanation: This command is prohibited because the target is in an incompatible mode. For example, an ITIMER request is not accepted when the system is power-on reset in LPAR mode.

System Programmer Response: This function cannot be performed in the present state of the receiver. Retry the request after the target mode status has changed.

08090001 Mode inconsistency: The requested function cannot be performed in the present state of the receiver.

Explanation: Acceptance of the command is prohibited because the target is in an incompatible mode. For example, an ITIMER request is not accepted when the system is power-on reset in LPAR mode.

System Programmer Response: None. This function cannot be performed in the present state of the receiver.

08090027 Mode inconsistency: The requested function cannot be performed in the present state of the receiver.

Explanation: The receiving Hardware Management Console is not in the correct state to automatically dial out using the attached modem.

System Programmer Response: Ensure the receiving Hardware Management Console is customized to use the auto-dial and RSF functions.

08090051 Mode inconsistency: The requested function cannot be performed in the present state of the receiver.

Explanation: Operations management control is not enabled.

System Programmer Response: Enable the system for automated operations and resend the request. Ensure that the Emergency Power Off switch is on.

080A000A Permission rejected: The receiver has denied an implicit or explicit request of the sender.

Explanation: A STATLEV request was rejected because it was not compatible with the status reporting values set in the receiver.

System Programmer Response: Correct the STATLEV value and resend the request.

080A000C Permission rejected: The receiver has denied an implicit or explicit request of the sender.

Explanation: A SETCLOCK request has failed because it required that a clock be set in a configuration where a dominant timing source has priority.

System Programmer Response: If the Sysplex Timer is the dominant timing source, then the SOURCE, TIME, UTCO, and OFFSET operands cannot be used in the command string. Remove these operands and resend the request.

080C0005 Procedure not supported: A procedure specified is not supported in the receiver.

Explanation: The command is not supported.

System Programmer Response: Resend the request using a supported command, if possible.

080C0007 **Procedure not supported: A procedure specified is not supported in the receiver.**

Explanation: A request for a function is supported by the receiver, but the resource identified in the request does not support that function. This function cannot be canceled.

System Programmer Response: None.

080F0001 **End-user not authorized: The requesting end-user does not have access to the requested resource.**

Explanation: Authorization checks have not been successfully passed.

System Programmer Response: Correct the command authorization-token and resend the request.

08120000 **Insufficient resource: The receiver cannot act on the request because of a temporary lack of resource.**

Explanation: System resources are temporarily busy.

System Programmer Response: Resend command if required.

08120011 **Insufficient resource: The receiver cannot act on the request because of a temporary lack of resource.**

Explanation: Insufficient storage is available to the target component to satisfy the request.

System Programmer Response: Resend command.

08120012 **Insufficient resource: The receiver cannot act on the request because of a temporary lack of resource.**

Explanation: A timed command was rejected because the OCF timed operations queue was full.

System Programmer Response: Cancel any unnecessary scheduled requests and resend the command.

08150001 **Function active: A request to activate an element or procedure was received, but the element or procedure was already active.**

Explanation: Unable to perform the command because the target CPC Subset or CPC Image is operational and the force operand has not indicated the override selection.

System Programmer Response: Put the system in the appropriate state and resend the command.

081A0000 **Request sequence error.**

Explanation: Unable to perform the command because the target partition is in the deactivated state.

System Programmer Response: Activate the logical partition, then resend the original request.

Hardware object status summary

Table 23 lists the status values for CPC and image objects provided by the z900 API. The status description was taken from the HMC online help because the API documentation does not provide this information. Note that the status numbers are displayed with the asynchronous report message AOFA0100.

Table 23. Status values for CPC and image objects provided by the z900 API

OPERATING	0001	**Image:** All of the image's processors are operating. **CPC:** All of the CPC's processors are operating.
NO POWER	0004	**CPC:** CPC power is off.
NOT OPERATING	0002	**Image:** None of the image's processors are operating, but the exact status of the processors vary. **CPC:** *If a power-on reset has not been performed*: The CPC's processors cannot operate until a power-on reset of the CPC is performed. *If a power-on reset was performed*: None of the CPC's processors are operating, but the exact status of the processors vary.

Table 23. Status values for CPC and image objects provided by the z900 API (continued)

NOT ACTIVATED	0008	**Image:** The image is defined in the CPC's current input/output (I/O) configuration, but is not activated.
EXCEPTIONS	0010	**Image:** At least one of the image's processors is operating, and at least one processor is not operating, but the exact status of the processors vary. **CPC:** At least one of the CPC's processors is operating, and at least one processor is not operating, but the exact status of the processors vary.
STATUS CHECK	0020	**Image:** The CPC is not communicating with the support element. The status of the image and its CPs cannot be determined. **CPC:** The CPC is not communicating with the support element.
POWERSAVE	0100	**CPC:** Utility power for the CPC failed, and one or more of its active control programs put the CPC in a power save state. The CPC is using only enough power from its alternate, temporary power source to preserve data for the control programs that put it in the power save state. **Image:** The image cannot operate until power for the CPC is restored.
LINK NOT ACTIVE	0080	**CPC:** The CPC's support element is not communicating with this HMC. The status of the CPC cannot be determined.
SERVICE	0040	**CPC:** A console operator enabled service status for the CPC (ordinarily done at the request of a service representative to allow providing service for the CPC).
SERIOUSALERT	0200	No explanation found on the HMCs.
ALERT	0400	No explanation found on the HMCs.
ENVALERT	0800	No explanation found on the HMCs.
SERVICE REQUIRED	1000	The next disruption will result in the CPC operating in degraded capacity, or it will fail to operate.
DEGRADED	2000	The CPC was found operating in a degraded state. Note that this status is set on specific hardware. Use the HW command GETSGDR to determine the reason.

Hardware object status summary

Glossary

This glossary defines technical terms and abbreviations used in z/OS Managed System Infrastructure for Operations documentation. If you do not find the term you are looking for, view *IBM Glossary of Computing Terms*, or *Tivoli*® *Glossary* located at:

`http://www.ibm.com/ibm/terminology`

`http://publib.boulder.ibm.com/tividd/glossary/termsmst04.htm`

A

authorized (message) receiver. The operator who is authorized to receive all the unsolicited and authorized messages that are not routed to a specific operator.

automation router task. The automation router task receives the unsolicited messages that are defined as automatable and initiates the automated response if the automation table contains an entry for this message.

autotask. An unattended operator station task that does not require a terminal or a logged-on user. Autotasks can run independently of VTAM and are typically used for automated console operations.

C

CDS. Couple Data Set.

CF. Coupling Facility.

CFRM. Coupling facility resource management.

CFRM policy. A set of *coupling facility* and *structure* definitions. The purpose of a CFRM policy is to specify which coupling facilities and structures are available in the sysplex. There can be several CFRM policies, but only one of these can be active at any given time. Activating a different CFRM policy is possible at runtime. The CFRM policies are contained in the CFRM *couple data set*.

connector. An application (component) that is connected to a *structure*.

couple data set. A data set that contains control information about the sysplex or a sysplex–related function such as *CFRM*. Every MVS system of a sysplex must have access to the couple data set of all functions that are implemented in the sysplex.

coupling facility. A logical partition that provides storage for data exchange between applications across the sysplex. The storage of a coupling facility is divided into areas that are called *structures*. Structures are identified by their name.

cross-system coupling facility (XCF). A component of MVS that supports cooperation between authorized programs running within a sysplex on the same or different members.

cross-system extended services (XES). A set of services that allow authorized applications or subsystems running in a sysplex to share data using a *coupling facility*.

D

data services task (DST). The msys for Operations subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

DST. Data Services Task.

duplexing. Maintaining two instances of a *structure* with the same name on different *coupling facilities* at the same time. Duplexing allows applications to maintain a backup version of their structures so that the structure is still available when one of its instances has failed.

E

EMCS. extended multiple console support.

L

logical partition. A logical processor complex within a physical processor complex that operates independently of the other logical partitions within that physical processor complex.

LPAR. Logically Partitioned Mode.

M

message automation table. A table that specifies the automated response to certain messages. The entries of the table determine which command is to be issued in response to the respective message, and which autotask is to execute the command.

message processing facility (MPF). An MVS facility that controls message display and message processing.

multisystem application. An application program that has various functions distributed across members of a sysplex.

O

operator station task (OST). The 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 msys for Operations.

P

policy. A set of sysplex-related definitions that is contained in a *CDS* and takes effect by being activated. Some CDSs can contain more than one policy. In this case only one policy can be active at a given time.

PPT. primary POI task.

primary POI task (PPT) . The subtask that processes all unsolicited messages that are received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor.

preference list. An ordered list of *coupling facilities* that is associated with a *structure*. The preference list specifies the coupling facilities on which the structure can be allocated. Usually, the system allocates the structure on the first coupling facility in the list that is not excluded by other requirements (for example, the size of the structure). The preference list is part of the structure definition in a *CFRM policy*.

R

RACF. Resource Access Control Facility.

rebuild. A process by which data from an initial instance of a *structure* is reconstructed in another structure instance with the same name on the same or another *coupling facility*.

There are two types of rebuild, plain rebuild and *duplexing*. With the first type, the new structure instance will replace the initial one, whereas duplexing will simply create a second instance.

A rebuild can be accomplished by two methods, user-managed rebuild and (from release 8 onwards) system-managed rebuild. With user-managed rebuild, the *connectors* are responsible for reconstructing the data, whereas this is done by *XES* in case of a system-managed rebuild.

S

SFM. Sysplex Failure Management

SFM policy. A set of system-related entries that specify how a failure of that system or its connections within the sysplex is to be handled. There can be several SFM policies, but only one of these can be active at any given time. The active SFM policy can be switched at runtime. The SFM policies are contained in the SFM *couple data set*

SMF. System Management Facility.

solicited message. A message which is sent in response to an operator command, and which has a specific destination, such as an msys for Operations operator. Contrast *unsolicited message*

structure. A storage area in a *coupling facility* that is associated with a name. Structures serve to exchange data between applications components across the sysplex. They are allocated when an applications component requires to be connected to the structure; when all connected components have terminated the connection normally, the structure is deallocated again except when it has been declared persistent by a *connector*.

Most properties of the structure are specified by the connecting application, for instance its purpose and internal organization, whether the structure is persistent, and whether *rebuild* is allowed. However, the application does not select the coupling facility on which the structure is allocated; this is done by *XES* based on a *preference list*. The application only knows the name of the structure.

sysplex. One or more MVS systems that reside on one ore more processors and are connected with each other so that programs in one of the systems can communicate with programs in another system. There are two types of sysplexes, basic and parallel. Parallel Sysplexes usually contain one or more *coupling facilities*.

sysplex failure management. An MVS component that manages system and connectivity failures within a sysplex according to a predefined policy. The SFM couple data set contains the active and eventual alternate policies.

system logger. An MVS component which enables applications to log data without needing to know how and where the data is stored, and which allows multisystem applications to use a common log.

System Management Facility (SMF). A standard feature of z/OS that collects and records a variety of system and job-related information.

T

task. (1) A process with certain properties that is defined to, and runs within, msys for Operations. (2)

Started task, an MVS procedure that is started with the START command of MVS. (3) A major unit in a processing sequence.

U

unsolicited message. A message that was not expected in response to an operator action. Contrast *solicited message*.

W

write-to-operator (WTO). A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

write-to-operator-with-reply (WTOR). A request to send a message to an operator at the z/OS operator console 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 z/OS supervisor program.

WTO. Write-to-Operator.

WTOR. Write-to-Operator-with-Reply.

X

XCF. Cross-system coupling facility

XCF group. A set of program instances or functions across the sysplex that a multisystem application defines to XCF. Members of an XCF group can communicate with each other by XCF services.

XES. Cross-system extended services.

Notices

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

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

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

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

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

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

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Deutschland
Informationssysteme GmbH
Department 3982

Pascalstrasse 100
70569 Stuttgart
Germany

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Programming interface information

This publication documents information that is NOT intended to be used as a Programming Interface of msys for Operations.

Trademarks

The following terms are trademarks or service marks of International Business Machines Corporation in the United States, or other countries, or both:

Extended Services
IBM
MVS/ESA
NetView (Tivoli Systems Inc.)
OS/390
Parallel Sysplex
PR/SM
RACF
Redbooks
RMF
S/390
Tivoli
VTAM
z/OS
z/VM
zSeries

Microsoft®, Windows®, Windows NT® and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.

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

Other company, product, and service names may be trademarks or service marks of others.

Index

Readers' Comments — We'd Like to Hear from You

z/OS and OS/390
Managed System Infrastructure for Operations
Setting Up and Using

Publication No. SC33-7968-06

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

SC33-7968-06

IBM®

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Information Development
Department 3248
Schoenaicher Strasse 220
71032 Boeblingen
Federal Republic of Germany

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in USA

SC33-7968-06