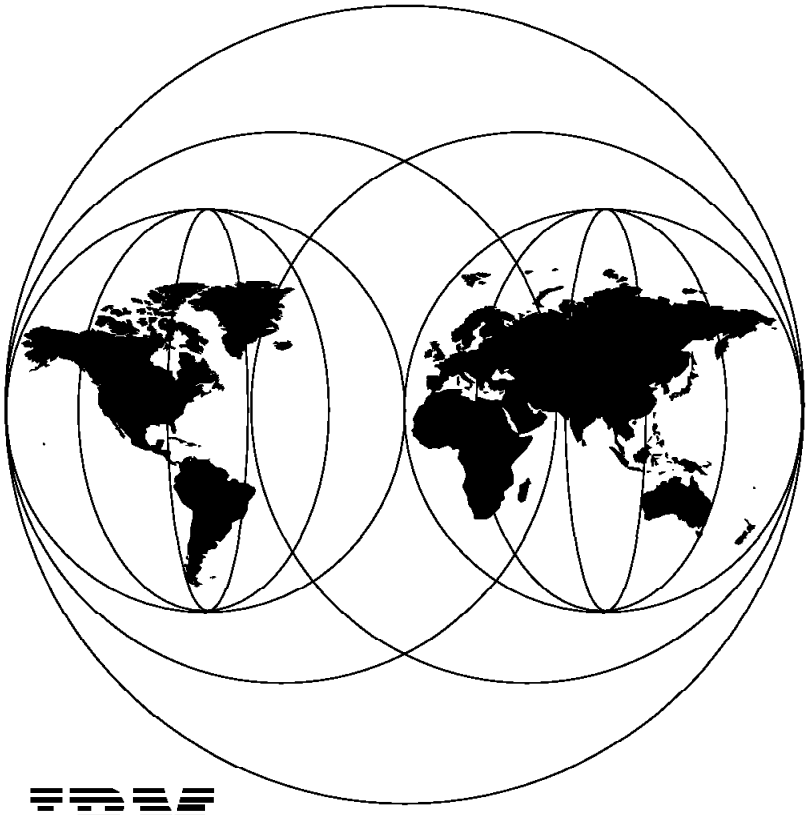
SG24-4778-00

RS/6000 SP: Problem Determination Guide

December 1996



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4778-00

RS/6000 SP: Problem Determination Guide

December 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special Notices” on page 281.

First Edition (December 1996)

This edition applies to Version 2, Release 1 of POWERparallel System Support Programs for use with the AIX 4.1.4

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

Contents

Figures

Tables

Preface

Problem determination and problem solving on the RS/6000 SP can be difficult because the malfunction may imply the involvement of several components within AIX and the POWERparallel System Support Programs.

This redbook describes each SP component and helps you understand how they are related to one another. It gives a comprehensive explanation of processes occurring within the SP system and provides an approach to diagnosis and problem solving.

For example, we describe how you can use the error log facility to handle AIX-related problems. For RS/6000 SP-specific problems, we show how you can use the PSSP component log files to solve them. All SP log information is located at /var/adm/SPlogs, and all PSSP components use this directory structure to store debugging information; the internal file structure and syntax are specific to each component.

The redbook also includes appendices with useful reference material about RS/6000 SP script files, the SDR structure, and how to change IP addresses and hostnames for the SP.

This redbook is a valuable tool for system administrators and other technical support personnel who deal with SP problems. It is also useful for those who want a more comprehensive understanding of RS/6000 SP components.

How This Redbook Is Organized

This redbook contains 295 pages. Most of the chapters can be read individually, except for Chapter 5, “System Partitioning,” for which Chapter 4, “The Switch” is a prerequisite.

This redbook is organized as follows:

- Chapter 1, “Overview”

  This chapter gives a brief overview of the main SP components. It explains how these components make up the system and which of them are specific to the RS/6000 SP. Experienced SP readers do not need to read this chapter.

- Chapter 2, “The Installation Process”

  This chapter covers the kind of problems that are frequently found when the system administrator is installing or customizing PSSP components. Typical problems with PSSP scripts, NIM components, and customization steps are covered here.

- Chapter 3, “Kerberos”

  Kerberos provides authentication services that allow certain distributed services within the SP system, or between it and other workstations, to secure control access to their facilities. This chapter explains the Kerberos concepts and how the components work together. It also covers the common problems that the Kerberos administrator faces, and gives some procedures to deal with them.

- Chapter 4, “The Switch”

  Most of the Switch-related problems require a clear understanding of the Switch components. This chapter explains those components and gives examples about configuration and topology files. Using these examples, the chapter explains how to handle Switch problems and how to interpret the Switch log files. This chapter also prepares the reader for Chapter 5, which requires a good understanding of the Switch components.

- Chapter 5, “System Partitioning”

  This chapter covers System Partitioning in a way that will enable the reader to understand how all the key components fit together, and give an understanding of how to carry out problem determination based on this information. It is essential that the reader fully understands the information laid out in Chapter 4 before beginning to read these details about System Partitioning.

- Chapter 6, “Error Logging”

  This chapter describes the error logging facilities provided by the PSSP components, and how these facilities interface with the standard AIX error logging “errlog” mechanism. This chapter also includes a description of how to customize the *errlog* to provide specific notifications that are not part of the standard set of notifications provided for the PSSP components.

- Chapter 7, “Isolating Problems on the SP System”

  The RS/6000 SP provides several tools that help to identify and isolate problems. In this chapter, these tools will be used, along with the Symptom Index in Chapter 3 of the *Diagnosis and Messages Guide*, GC23-3899 to help to isolate problems. Once the problems are isolated, the procedures explained in the previous chapters can be applied.

- Chapter 8, “Producing a System Dump”

  AIX allows you to generate Memory System Images, called *dumps*, that can be analyzed at a later time. Sometimes, under error conditions, these *dumps* are generated automatically by the operating system. This facility is present in the Control Workstation and in the nodes. This chapter explains how to generate system dumps and how to handle them using standard SP tools.

- Chapter 9, “User and Services Management”

  This chapter briefly describes some of the components of User and Service Management, such as File Collection, Amd, Print Services, and NTP. It also describes how to diagnose and handle common problems with these components.

- Appendix A, “RS/6000 SP Script Files”

  Many of the SP tasks are carried out by script files. This appendix provides flow charts for the main PSSP scripts, which can be used as a reference for problem determination.

- Appendix B, “The SDR Structure”

  The SDR is the core of the SP configuration data management. The structure consists of files, processes, and commands that allow you to retrieve information. Its structure is shown in this appendix as a reference for those who want to know more about its components and how they are interrelated.

- Appendix C, “IP Address and Hostname Changes for the SP”

  The IP and Hostname changing on the nodes or on the Control Workstation is one of the most difficult tasks faced by the administrator. This appendix provides a procedure that allows you to perform painless and successful IP and Hostname changing by giving some advice and an easy-to-follow procedure.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Marcelo R. Barrios is an SP Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of RS/6000 SP. Before joining the ITSO one year ago, Mr. Barrios worked as an assistant professor in the Electronic Department of Santa Maria University, Valparaiso, Chile. In 1993, he joined to IBM as a Marketing Specialist in the RS/6000 Unit, IBM Chile.

Richard Wagner is a AIX Software Specialist in the RS/6000 Software Support Center Mainz, IBM Germany. He joined IBM in 1989 and worked in RS/6000 SP support supporting the RS/6000 BOS software. During the last 10 months he has focused on the RS/6000 SP products. His expertise has been gained by working closely with customers resolving installation and post-installation issues. He has written extensively on the SP Switch. A special “thanks” is extended to Richards′s colleagues in the AIX Support Center, Mainz, for covering the work while Richard was writing the redbook.

Andy Hoiles is a Senior AIX Systems Specialist in the RS/6000 Support Center, IBM UK Ltd. He joined IBM in 1989 and began supporting the RS/6000 software in 1992. During the last 18 months he has focused on RS/6000 SP products and his expertise has been gained by working closely with customers resolving installation and post-installation issues.

Sunil Jain is an Advisory IT Specialist in Sydney, Australia. He is a certified AIX Support Professional and CNE. He has 18 years experience in the computer industry. He holds a Post Graduate (M.Sc) degree in Operations Research from University of Delhi. His area of expertise includes AIX (on RS/6000 and SP), LAN/WAN and communications/networking products. He has written extensively on LAN backup/recovery, EDI Implementation and SP Problem Determination.

Tony Schlee is an Advisory Systems Engineer at IBM South Africa. He has 8 years UNIX experience. He has worked for IBM since 1992 with AIX on the RS/6000.

Thanks to the following people for their invaluable contributions to this project:

Endy Chiakpo
International Technical Support Organization, Poughkeepsie Center

John Lobbes
RS/6000 SP Support Center, Poughkeepsie

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Overview

The flexibility of the RS/6000 SP plays a very important role when the customer has to choose between a conventional RISC machine, such as a Uniprocessor or Symmetric Multiprocessor, and a Massively Parallel Processor. However, this flexibility works against the situation if the customer tries to manage it as a conventional system.

We have to understand the philosophy behind the system to realize that there are many features which make the RS/6000 SP unique. And, although it runs AIX like any other RS/6000 machine, it has special pieces of hardware and software code that need special attention to support and manage them.

1.1 RS/6000 SP: Hardware and Software

Although RS/6000 SP is built with standard AIX and RS/6000 parts, it has its own hardware components and special software to make managing it easier.

There are three hardware components that make the RS/6000 SP different from conventional RISC machines:

- The Control Workstation
- The Frame
- The High Performance Switch

These components are integrated into the RS/6000 SP environment by the POWERparallel System Support Programs.

1.1.1 The Control Workstation

The Control Workstation (CW) is the console for the RS/6000 SP, but the CW is not *merely* a console. It allows you to manage the system as if it was a single unit, regardless how many nodes you have[1]. However, you will not get the same functionality you have in a standard RS/6000 console, because you are managing a Parallel System, and therefore, you are managing multiple systems at once.

The CW is a standard RS/6000 machine, but not all the models are supported as Control Workstations.

Note: See *POWER Parallel Service Bulletin N° 11* for more information about using supported RS/6000 models as Control Workstations.

Usually, the CW has at least two connections to the frame; this is shown in Figure 1 on page 2.

[1] This ability has often been referenced as ***Single System Image***.

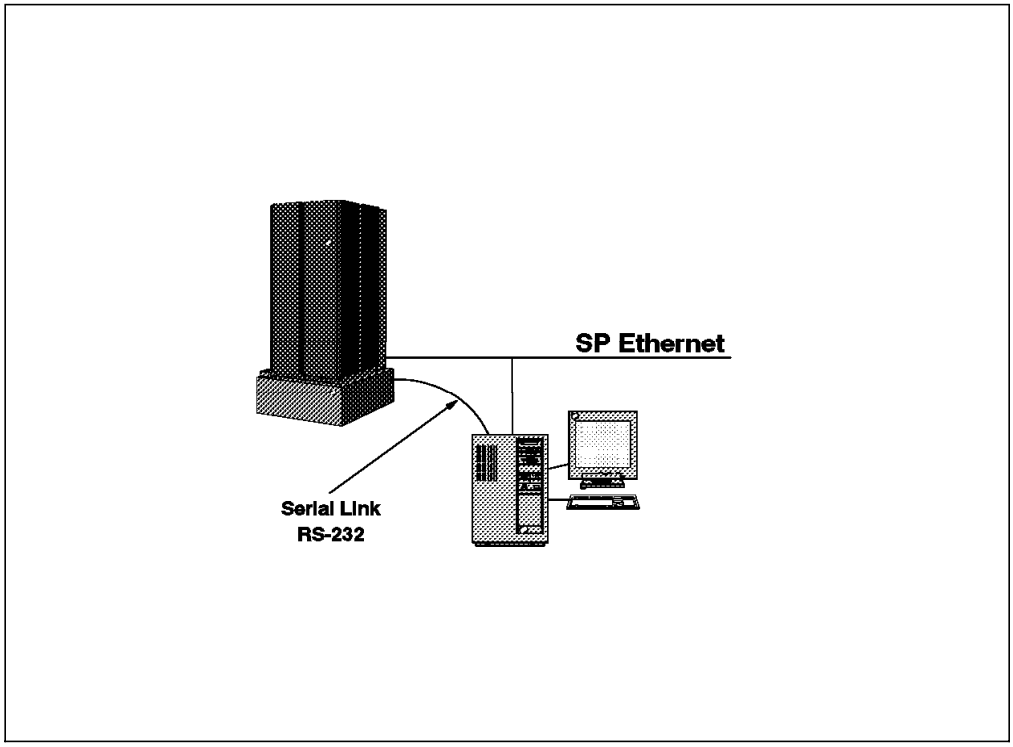


Figure 1. Control Workstation Connected to the SP Frame

The serial link connects the CW with the Supervisor Card into the Frame. In this way, the CW can manage and monitor every major hardware event produced either by the Frame itself or by the nodes. Each node has a Node Supervisor Card which is connected to the Frame Supervisor Card, as shown in Figure 2 on page 3. The new Supervisor Card is able to connect a second CW as a backup Control Workstation by using the HACWS software option for the POWERparallel System Support Programs.

Note: More information about HACWS can be found in *High Availability on SP*, SG24-4742.

The Ethernet network (called SP Ethernet) connects each node and the CW. Usually this network is dedicated to SP traffic only, therefore you should install a different network adapter to pass user applications traffic.

1.1.2 The Frame

All the hardware information flows from the Frame to the CW through the Serial Link. In this way, the CW is able to detect hardware failures in the nodes or in the Frame, and take actions over hardware components.

However, the CW uses the Ethernet to recognize when the nodes are “alive,” not just powered on, but up and running. Communications with the nodes is handled by the hardware daemon (called hardmon), which sends small packages called *heartbeat packages* through the TCP/IP stack to get in contact with the nodes. When the nodes are up and running, the hardware monitor sets the *host responds* variable from the *SDR* to “1” for each node.

At this time, the hardware monitor can use only the SP Ethernet to send heartbeat packages, which reduces this network to a single point of failure. But

in future releases, the SP will be able to send heartbeat packages over any existing network that connects the nodes and the CW.

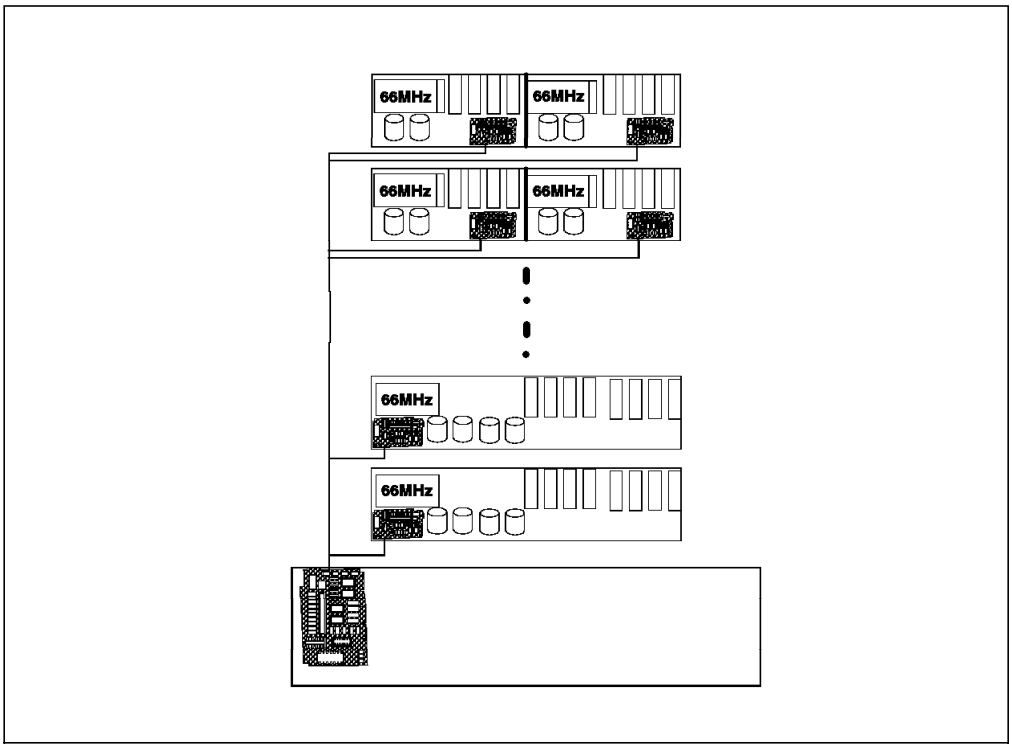


Figure 2. Supervisor Cards

The Frame contains redundant power supplies; if one power supply fails, another takes over. The Frame is also designed for concurrent maintenance; each node can be removed and repaired without interrupting operations on the other nodes.

1.1.3 The High Performance Switch

The High Performance Switch is one of the most unique pieces of hardware and software developed for the RS/6000 SP. The hardware portion of the switch provides a low latency and high bandwidth of communication between nodes. Each node is connected to the High Performance Switch by the Switch Adapter, as shown in Figure 3 on page 4.

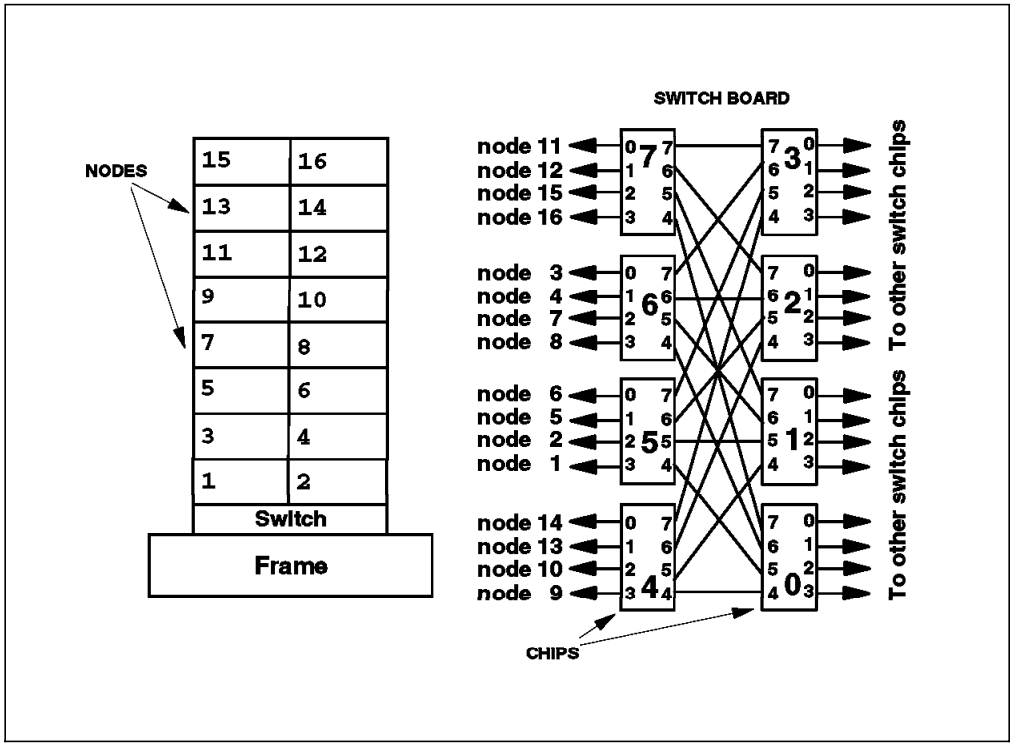


Figure 3. The High Performance Switch

As you can see in Figure 3, each node is connected to one chip port, and there are eight chips per switch board. Each chip has eight ports, four external ports connecting the nodes, and four internal ports connected to the mirror chips.

This topology gives each node four different routes to communicate with another node. Therefore, each node has four possible ways to reach another node, and it could use all of the routes together to have a very high bandwidth.

The software portion of the switch provides the *device drivers*, the *processes* and the *commands* needed to use and manage the switch. These components will be covered later on this book.

The Switch has the ability to partition, which creates separate and non-disrupting environments. In this way, each partition can be seen and managed as a separate SP system. This topic will be covered in more detail in the next chapter.

1.2 Problem Determination

Basically, there are two kinds of problems encountered on the RS/6000 SP: those related to AIX and those related to the POWERparallel System Support Programs. For those related to AIX, the approach to solve them is the standard AIX way, which means you can use *errlog* and *trace* facilities to find the problem.

For those problems specific to the RS/6000 SP, each PSSP component usually provides log files and tools to solve them. All the log information is located on */var/adm/SPlogs*. The directory structure is shown in Figure 4 on page 5.

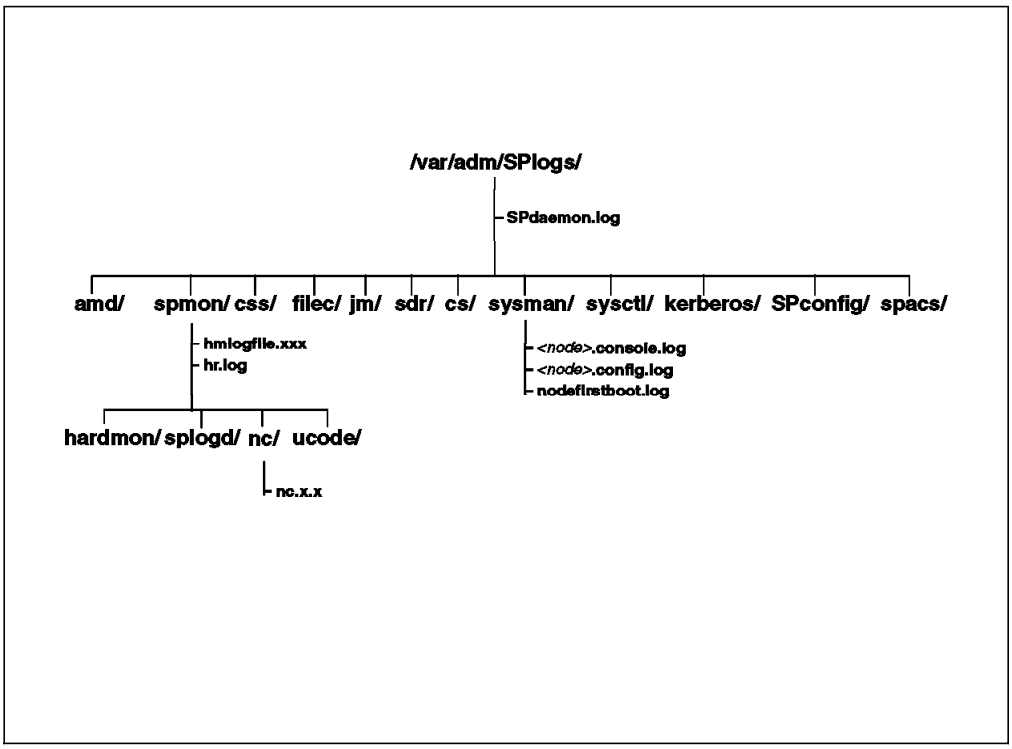


Figure 4. SP Error Log Structure

All the PSSP components use this directory structure to write their debugging information. The internal file structure and the syntax is specific to each component.

When you have problems on the RS/6000 SP, it is sometimes difficult to determine the source of the problem, especially when the problem or malfunctioning involves several components within AIX and PSSP. One of the most difficult things to do is to isolate the problem to one component. Most of the problems have clear symptoms that point in the right direction, but sometimes there are problems with symptoms that are not clear enough to signal which component is causing it. For instance, if a command like this:

```
# dsh -w sp21n01
```

fails and says that it could not be authenticated on node sp21n01 due to a kerberos ticket problem, the offensive component is clear. But if you enter this command:

```
# SDRGetObjects SP
```

and you get nothing, the problem could imply several components, including hardware failures.

Problem determination and problem solving on SP are difficult tasks to perform without the right documentation and procedures. Even good documentation is not always enough to fix a problem. In order to be successful, it is necessary to understand all of the SP components and realize how they are related to one another.

This book addresses that problem, by giving a comprehensive explanation to the processes occurring within the SP components and providing an approach to solve SP problems. Each component has its own chapter, and all of them are self-contained. The appendixes included cover those extensive parts that are

not considered suitable to include in a chapter but are useful as reference material.

Chapter 2. The Installation Process

Installing the RS/6000 SP system includes installing both hardware and software. Before RS/6000 SP system hardware and software are installed, a detailed plan should be created. For details of RS/6000 SP site planning, refer to *RS/6000 SP: Site Planning*, GC23-3905.

Details of RS/6000 SP hardware installation are covered in *Volume 1: Installation and CE Operations*, GC23-3903.

Details of RS/6000 SP software planning and installation procedures are covered in *RS/6000 SP: System Planning*, GC23-3902 and in *RS/6000 SP: Installation Guide*, GC23-3898.

References to other RS/6000 SP hardware and software manuals can be found in *IBM RS/6000 Scalable POWERparallel Systems Library*, GC23-3868.

For details on the installation of RS/6000 SP, refer to *RS/6000 SP: Installation Guide*, GC23-3898, Chapter 2, “Installing and Configuring the RS/6000 SP System.” For important tips, in an easy-to-use cookbook format, see *RS/6000 Scalable POWERparallel Systems: PSSP Version 2 Technical Presentation*, SG24-4542, Chapter 3, “Installing and Configuring the SP System.”

For the purpose of this book, we assume that detailed site planning has been carried out, RS/6000 SP hardware has been correctly installed, and RS/6000 SP software planning has already completed. We are now ready to install RS/6000 SP software. This chapter deals with explaining checkpoints at various stages of the PSSP software install process and provides tips to resolve general problems.

2.1 Major Steps in Installing and Configuring RS/6000 SP Software

This chapter follows *RS/6000 SP: Installation Guide*, GC23-3898, Chapter 2. “Installing and Configuring the RS/6000 SP System,” which explains each of the installation steps in detail. Therefore, this chapter should be read if you need tips to solve or avoid installation problems, or if you need to understand various checkpoints in the install process. Following are the major checkpoints in the installation of the RS/6000 SP system.

1. Setting up the Control Workstation
2. Authentication services
3. The install_cw script
4. The setup_server script
5. Installing the nodes
6. Post-install customizations

The following section discusses the diagnostics for each of these steps.

2.2 Prepare the Control Workstation

This section is covered in steps 0 through 11, in the *Installation Guide* and in sections 3.1, 3.2.1, and 3.2.2 in *RS/6000 SP: PSSP Version 2 Technical Presentation*, SG24-4542.

Completing steps 0 through 11, means that the following have been successfully achieved:

1. AIX has been successfully installed with paths properly defined in *.profile* for the administrator and other users.
2. All network connections have been made. Hostnames, IP addresses, and netmasks have been correctly defined. All IP addresses are able to ping successfully.
3. RS-232 control lines have been configured on the Control Workstation for the frame.
4. Space for SP data has been defined (/spdata filesystem).
5. AIX 4.1.4 LPP images have been copied to /spdata/sys1/install/lppsource.
6. PSSP install images have been copied to /spdata/sys1/install/pssplpp and renamed as pssp.installp, and inutoc script has been run.
7. Basic AIX mksysb image *bos.obj.ssp.41* has been installed under /spdata/sys1/install/images.
8. PSSP software has been installed on the Control Workstation.
9. The required PTF set has been installed.

Any problems that you experience to this point will be AIX software- or hardware-related. (You have not yet encountered &sp. software.) Refer to AIX documentation for resolution tips. The following sections cover some general problems that occur while setting up the Control Workstation and how to resolve them. Additional information has also been provided to prevent problems.

2.2.1 PSSP Paths

Incorrect paths (or the absence of correct paths) can result in various problems and wasted time during and after installation. Immediately after installing your Control Workstation with AIX and LPPs, create a .profile file in your root directory. Figure 5 on page 9 is an example of a .profile file.

```
PATH=/usr/lpp/ssp/rcmd/bin:
PATH=$PATH:/usr/lpp/ssp/bin:          \
    /usr/lpp/ssp/kerberos/bin:/usr/local
PATH=$PATH:/usr/bin:/etc/:/usr/sbin:/usr/ucb
PATH=$PATH:/usr/dt/bin:/usr/lpp/X11/bin:/sbin:
PATH=$PATH:$HOME/bin:/sbin:/bin
MANPATH=/usr/lpp/ssp/man:/u/loadl/man:/usr/man
ENV=/.kshrc
export PATH MANPATH ENV

if [ ! "$DT" ]
then
   if [ -s "$MAIL" ]
   then echo "$MAILMSG"
   fi
fi
```

Figure 5. Sample of a Portion of .profile File

AIX 4.1.4 comes with Common Desktop Environment (CDE), and many customers decide to use this. AIX provides a sample dtprofile under /usr/dt/config/sys.dtprofile.

/usr/dt/config/sys.dtprofile is a factory-default file and will be unconditionally overwritten upon subsequent installation. Before making changes to the file, copy it to the configuration directory /etc/dt/config.

The sys.dtprofile file is copied to $HOME/.dtprofile the first time a user logs into the desktop. Any lines in sys.dtprofile located between “SYSPROFILE COMMENT START” and “SYSPROFILE COMMENT END” are filtered out during the copy.

Your $HOME/.dtprofile is read each time you log in to the Common Desktop Environment (CDE) and is the place to set or override desktop environment variables for your session. Environment variables set in $HOME/.dtprofile are made available to all applications on the desktop. The desktop will accept either sh or ksh syntax for the commands in $HOME/.dtprofile.

By default, the desktop does not read your standard $HOME/.profile or $HOME/.login files. This can be changed by uncommenting the DTSOURCEPROFILE variable assignment at the end of this file. The desktop reads .profile if your $SHELL is “sh” or “ksh,” or .login if your $SHELL is “csh.”

Here is an example of a sample .dtprofile.

```
DTSOURCEPROFILE=true
```

For more details, read the text under /usr/dt/config/sys.dtprofile.

2.2.2 AIX Software Components

The RS/6000 you are using as the Control Workstation (CWS) for RS/6000 SP System must have the following software installed:

- AIX Version 4.1 Base Operating System
- TCP/IP
- NFS
- NIM file sets

Figure 6 shows the `lslpp` command used to check the base operating system runtime environment components that are installed on your Control Workstation.

```
# lslpp -l bos.rte
  Fileset                 Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.rte               4.1.4.0  COMMITTED  Base Operating System Runtime

Path: /etc/objrepos
  bos.rte               4.1.4.0  COMMITTED  Base Operating System Runtime
```

Figure 6. Minimum Required BOS and Runtime Environment Components

Figure 7 shows the command to check Network Install Manager (NIM) components that are installed on your Control Workstation. If it does not list all of the required components, then install the missing NIM components from the AIX install media before proceeding further.

```
# lslpp -l "bos.sysmgt.nim.*"
  Fileset                   Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.sysmgt.nim.client    4.1.4.0  COMMITTED  Network Install Manager -
                                                Client Tools
  bos.sysmgt.nim.master    4.1.4.0  COMMITTED  Network Install Manager -
                                                Master Tools
  bos.sysmgt.nim.spot      4.1.4.0  COMMITTED  Network Install Manager -
                                                SPOT
Path: /etc/objrepos
  bos.sysmgt.nim.client    4.1.4.0  COMMITTED  Network Install Manager -
                                                Client Tools
```

Figure 7. Minimum Required Components of Network Install Manager (NIM)

Figure 8 on page 11 shows the command used to check networking bundles that are installed on your Control Workstation. If it does not list all of the required components, then first install the missing components from the AIX install media before proceeding further.

```
# lslpp -l "bos.net.*"
  Fileset                      Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.net.nfs.client       4.1.4.0  COMMITTED  Network File System Client
  bos.net.nfs.server       4.1.0.0  COMMITTED  Network File System Server
  bos.net.tcp.client       4.1.4.0  COMMITTED  TCP/IP Client Support
  bos.net.tcp.server       4.1.4.0  COMMITTED  TCP/IP Server
  bos.net.tcp.smit         4.1.4.0  COMMITTED  TCP/IP SMIT Support

Path: /etc/objrepos
  bos.net.nfs.client       4.1.4.0  COMMITTED  Network File System Client
  bos.net.tcp.client       4.1.4.0  COMMITTED  TCP/IP Client Support
  bos.net.tcp.server       4.1.4.0  COMMITTED  TCP/IP Server
```

Figure 8. Minimum Required Components of Base Networking Software

2.2.3 Disk Space Requirements

Check that you have enough disk space on the Control Workstation. (You need about 2 to 4 GB of free disk space).

Figure 9 shows the command to check free physical partitions (PPs) on rootvg. In this example, there are 1852 free PPs (7408 megabytes) available under the rootvg volume group on the Control Workstation.

```
# lsvg rootvg
VOLUME GROUP:   rootvg             VG IDENTIFIER:  000081007414db91
VG STATE:       active             PP SIZE:        4 megabyte(s)
VG PERMISSION:  read/write         TOTAL PPs:      2166 (8664 megabytes)
MAX LVs:        256                FREE PPs:       1852 (7408 megabytes)
LVs:            12                 USED PPs:       314 (1256 megabytes)
OPEN LVs:       11                 QUORUM:         3
TOTAL PVs:      5                  VG DESCRIPTORS: 5
STALE PVs:      0                  STALE PPs       0
ACTIVE PVs:     5                  AUTO ON:        yes
```

Figure 9. Output of lsvg rootvg Command to Check Disk Space

2.2.4 RS-232 Control Lines Diagnostics

Each frame in the RS/6000 SP system requires a serial port on the Control Workstation to accommodate an RS-232 connection between them. As there is only one frame in the testing environment, only one RS-232 line is needed to be setup on the Control Workstation. This could be achieved by the `smit tty` or the `mkdev` command. SMIT is the recommended method. Here is the command that SMIT constructs:

```
# mkdev -c tty  -t 'tty' -s 'rs232' -p 'sa0' -w 's1'
```

The hardmon daemon that executes on the Control Workstation uses the RS-232 line to each frame to poll the frame to check the state of the hardware within the frame. If the RS-232 line is not configured or the RS-232 connection is missing, then hardmon will not be able to poll the frame. Accordingly, the client commands `spmon` and `hmmon` will not be able to get the current or changing state

of the hardware. The splogd daemon, which is also a client of the hardware monitor daemon, will not be able to log any changes in hardware state.

2.2.5 Changing IP Addresses and Hostnames

The Control Workstation, in our examples, has a Token Ring and an Ethernet adapter card. Each of these two adapter cards is configured using **smit mktcpip**. SMIT built up the following commands:

```
# /usr/sbin/mktcpip -h'sp2tr0' -a'9.12.0.37' -m'255.255.255.0' \
  -i'tr0' -g'9.12.0.32' -r'16' -s''
# /usr/sbin/mktcpip -h'sp2cw0' -a'9.12.20.99' -m'255.255.255.0' \
  -i'en0' -t'bnc' -s''
```

tr0 is configured first, followed by *en0*. This is done to make *en0* the primary interface to the Control Workstation with hostname *sp2cw0*.

After configuring the *en0* and *tr0* interfaces, IP addresses and host names for nodes and Ethernet interfaces on the switch are defined. This is done based on the worksheets which were filled in after proper planning. AIX stores all host names and IP addresses in an ASCII file host under directory /etc. Figure 10 on page 13 shows the /etc/hosts file.

```
127.0.0.1       loopback localhost
# CWS Ethernet/TR interfaces
9.12.20.99      sp2cw0 sp2en0
9.12.0.37       sp2tr0
# sp2 nodes
9.12.20.1       sp2n01
9.12.20.2       sp2n02
9.12.20.3       sp2n03
9.12.20.4       sp2n04
9.12.20.5       sp2n05
9.12.20.6       sp2n06
9.12.20.7       sp2n07
9.12.20.8       sp2n08
9.12.20.9       sp2n09
9.12.20.10      sp2n10
9.12.20.11      sp2n11
9.12.20.12      sp2n12
9.12.20.13      sp2n13
9.12.20.14      sp2n14
9.12.20.15      sp2n15
9.12.20.16      sp2n16
# sp2 switches
9.12.6.1        sp2sw01
9.12.6.2        sp2sw02
9.12.6.3        sp2sw03
9.12.6.4        sp2sw04
9.12.6.5        sp2sw05
9.12.6.6        sp2sw06
9.12.6.7        sp2sw07
9.12.6.8        sp2sw08
9.12.6.9        sp2sw09
9.12.6.10       sp2sw10
9.12.6.11       sp2sw11
9.12.6.12       sp2sw12
9.12.6.13       sp2sw13
9.12.6.14       sp2sw14
9.12.6.15       sp2sw15
9.12.6.16       sp2sw16
```

Figure 10. /etc/hosts File on the Control Workstation

2.2.5.1 Verify Control Workstation Interfaces

Verify that the Ethernet adapter has been properly configured by pinging its IP address. A successful ping means that the adapter has been correctly configured. If there are problems pinging the IP address of the Ethernet adapter, then check for conflicting IP addresses, improper netmask values, or hardware problems. /etc/hosts may be a good place to look for conflicting IP addresses or host names. Refer to AIX TCP/IP documentation for details on how to resolve name resolution or routing problems. Figure 11 on page 14 shows a sample output of a successful ping.

```
/> ping 9.12.20.99 -c 1
PING 9.12.20.99: (9.12.20.99): 0 data bytes
8 bytes from 9.12.20.99: icmp_seq=0 ttl=255

----9.12.20.99 PING Statistics----
1 packets transmitted, 1 packets received, 0% packet loss
```

Figure 11. Output of a Successful Ping

It is important to plan the IP addressing scheme and hostnames before starting, because changing them later after configuring your network could become a major problem. There are procedures to change IP addresses and hostnames if required, but this adds extra steps and complications, in some cases. Refer to Appendix C, "IP Address and Hostname Changes for the SP" on page 273 for a procedure to change IP addresses or hostnames for the SP nodes and Control Workstation after initial installation and configuration. This procedure has been tested in the lab environment based on the /etc/hosts file for name resolution. This procedure has not been tested in other environments.

2.2.6 Maximum Number of Processes

As the AIX 4.1 default, the maximum number of processes allowed per user is limited to 40. (This limit does not apply to a root user.) The possible values range from 40 to 131072. Any increase in this number will take effect immediately. Any decrease in this number will not become effective until the next system boot.

Installing your RS/6000 SP system as a root user means that your system will not affected by this limit. However, it is a good idea to increase this value, based on your application requirements now, to cater to later requirements. This will ensure that non-root users running various applications later during production environment will not be limited by the default limit of 40 processes per user.

As an example, the following command changes the maximum number of processes allowed per user to 256. (Remember, this value needs to be set up to cater to your environment.)

```
# chdev -l sys0 -a maxuproc='256'
```

2.2.7 Number of Licensed Users

Check and rectify the number of licenses that you have bought for AIX 4.1 from IBM. By default, you will have two users licensed on your Control Workstation. Any changes will be effective after system reboot. Use smit fastpath **smit chlicense** or command line syntax:.

The following example shows how you can change the number of licensed users to 64. This is only an example, and the actual value depends on what you have bought from IBM.

```
# chlicense -u'64'
```

2.2.8 Tunable Values

After the initial installation of RS/6000 SP system, the network tunable values are set to AIX 4.1 defaults. Change these values to the optimum values for SP systems. This is achieved by setting *no -o* commands in tuning.cust file under *tftpboot* directory. When the Control Workstation is rebooted, rc.sp script will check for the existence of /tftpboot/tuning.cust file and run it if it exists. Refer to 2.11, “Node Customization Problems” on page 69 for details of post-installation customization steps.

Note: /etc/rc.sp does not exist yet. It will be created after the installation of PSSP software.

2.2.9 /spdata Directory Structure

You need to create a filesystem for SP data. You may create this filesystem under *rootvg* or another volume group. IBM recommends a separate volume group for this purpose. Here are some simple reasons for this recommendation:

- Independence from root volume group
- Speedy mksysb of the rootvg
- Performance
- Ease of managing programs and data in separate volume groups

Figure 12 shows the directory structure of */spdata* filesystem.

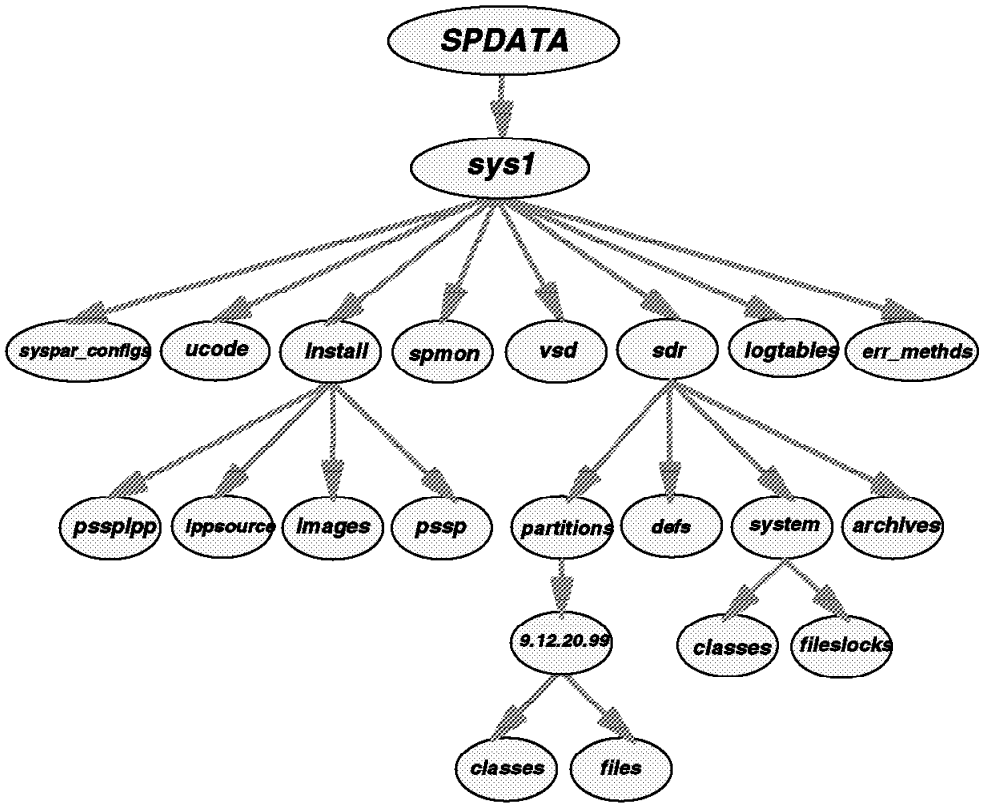


Figure 12. /spdata Directory Structure

Here is the brief description of the main directories of */spdata* filesystem:

Directories	Description
/spdata/sys1	Main directory for SDR, SP monitor, log management, partition layout files and installation.
/spdata/sys1/install	Main directory for PSSP installation.
/spdata/sys1/install/lppsource	Location of required AIX 4.1 file sets.
/spdata/sys1/install/images	Location of AIX system backup (mksysb) images.
/spdata/sys1/install/pssp	Location of NIM configuration data files.
/spdata/sys1/install/pssplpp	Location of all PSSP and SP system file sets.
/spdata/sys1/sdr	Location of the SDR files.
/spdata/sys1/sdr/archives	In this directory are stored the archives of the SDR.
/spdata/sys1/sdr/defs	Location of the header files for SDR classes.
/spdata/sys1/sdr/partitions	Location of the partition classes. There is one subdirectory for each partition.
/spdata/sys1/sdr/system	Location of the systemwide SDR classes.
/spdata/sys1/syspar_configs	Location of the topology files.
/spdata/sys1/ucode	Location of the microcode.
/spdata/sys1/spmon	Location of the System Monitor files (hmacls and hmthresholds).
/spdata/sys1/vsd	Location of the VSD files.
/spdata/sys1/logtables	Location of samples of service collection table for different PSSP components.
/spdata/sys1/err_methods	Location of the error notification methods.

Table 1. /spdata Filesystem Directories

Directories under */spdata/sys1/install* are manually created before the PSSP install process is started. The installation process will require *lppsource, images, pssp* and *pssplpp* directories under /spdata/sys1/install. Any deviation from these names will result in installation failure. Also ensure that these directories have the permission of rwxr-sr-x.

Other directories and information under them is created by the install process.

2.2.10 Install PSSP on the Control Workstation

Table 2 on page 17 provides a brief overview of various components of PSSP software, what the minimum required options are, and also what is recommended by IBM to be installed. Our recommendations will be overridden by your requirements.

Component	Option	Minimum	Recommended	Switch	Partition	Parallel
Authentication Server	ssp.authent	X	X			
Monitoring the SP	ssp.basic	X	X			
SP User Commands	ssp.client	X	X			
Switch Device Driver	ssp.css		X	X		X
Documentation	ssp.docs		X			
System Monitor GUI	ssp.gui	X	X			
Resource Manager	ssp.jm					X
Public Domain S/W	ssp.public					
Sysctl	ssp.sysctl	X	X			
SP Management Tools	ssp.sysman		X			
Partitioning Files	ssp.top		X		X	

Table 2. Components of the PSSP Install Image

2.2.10.1 Minimal PSSP Installation

For a minimum installation the following components are required:

- Code for installing and monitoring the SP system (**ssp.basic**)
- SP user commands (**sp.client**)
- System monitor GUI (**ssp.gui**)
- Sysctl (**ssp.sysctl**)

Other components may be required based on your environment.

If you are using MIT Version 4 or AFS authentication services, then *ssp.authent* is not required. For an RS/6000 SP authentication server, you must have *ssp.authent*.

If you have a High Performance Switch or an SP Switch, then you have to install the Switch Device Driver on your nodes. It is part of *ssp.css*.

If you plan to partition your SP system, then you need the predefined partition files included under *ssp.top*.

If you are using parallel applications, then you need Resource Manager *(ssp.jm)*.

You may leave out *ssp.public code*.

If you are not sure of what you require, then you may install all options of the PSSP install image on your workstation.

Running *smit install_latest* creates the following command:

```
/usr/lib/instl/sm_inst installp_cmd -T iems -L -q -a -d  \
'/spda/install/pssplpp/pssp.installp' -o 'all_licensed'  \
'-c' '-N' '-g' '-X'
```

2.2.11 PSSP 1.2

The following are some PSSP 1.2 considerations:

- AIX Version 3.2.5

  AIX Version 3.2.5 provides support for PSSP 1.2. Customers should upgrade to AIX Version 4.1 in order to get the benefits of enhanced features of PSSP 2.1. Customers can then have partitions for AIX Version 3.2.5 running PSSP 2.1, while testing their applications under AIX Version 4.1 partitions.

  Note: Customers need to fully understand what is involved in upgrading their Control Workstation to AIX Version 4.1. They also need to create AIX 3.2.5-based partitions before actually starting the upgrade.

- First Implementation of PSSP 1.2

  PSSP 1.2 was IBM's first implementation of POWERparallel System Support Programs (PSSP) software. This level of PSSP requires AIX Version 3.2.5. PSSP 1.2 is still supported by IBM developers and Support Centers. PTFset23 is the latest level of PTF available for PSSP 1.2. PTFset23 should be installed, because it resolves various known defects and also provides some enhancements.

- Field Experiences and Demands

  Based on customers' experiences in the field with PSSP 1.2 and customers' requests for enhanced features, PSSP Release 2 was made available in 1995. The new software fixes the defects that were identified under PSSP 1.2.

2.2.12 PSSP 2.1

The following are some PSSP 2.1 considerations:

- AIX 4.1.4

  PSSP 2.1 allows the SP system to exploit the benefits of AIX Version 4.1 and to prepare for future hardware and software technological advances.

- PSSP 2.1

  This is the latest PSSP code available. PSSP 2.1 has provided major enhancements and extra features over PSSP 1.2. Some of these are discussed here.

- Enhancements
  - System partitioning

System partitioning supports several system partitions running AIX Version 3.2.5 or AIX Version 4.1 to run as logical systems within a single SP system. This provides the ability to test new software, isolate workloads to dedicated resources, and run different applications on different partitions as required by different segments of the business.

- Node isolation from the switch network

  Node isolation provides a mechanism to isolate nodes from the switch fabric for repair, replacement, or reboot, without affecting end users, applications, or switch traffic on the remainder of the SP system or system partition.

- Installation support

  Improved installation support is provided by the use of the AIX Version 4 Network Installation Management (NIM).

- Improved log management

  Log management provides a single point of control to assist customers in configuring, archiving, and maintaining various system logs on individual nodes. Customers can also specify additional logs that may be used by their applications.

- Enhanced security features

  PSSP 2.1 increased the maximum ticket life to 30 days for Kerberos, and enhancements to address any host by any of its host names.

- VSD performance improvements

  With standard VSD, database applications could only stripe data across multiple disks associated with one node. With Hashed Shared Disk (HSD), the capability to stripe across nodes has been added, returning more data to the application faster than serial reads.

- Engineering and scientific computing

  The following software products have been upgraded with new functions to extend support for engineering and scientific computing:

  - IBM Parallel Environment for AIX Version 2.1 provides support for parallel applications on AIX Version 4.1.
  - IBM Parallel ESSL for AIX 4.1.4 improves performance of engineering and scientific applications on the RS/6000 SP Systems.
  - IBM PVMe for AIX supports parallel execution of applications on AIX version 4.1.4.

2.2.13 Changes from PSSP 1.2 to PSSP 2.1

The following are the changes made in going from AIX Version 3.2.5 to AIX Version 4.1:

- /usr client and /usr server

  /usr client and /usr server, which were supported under AIX Version 3.2.5, are not supported under the new version.

- Primary dump device

  Primary dump space, which was defined as /dev/hd7 under AIX Version 3.2.5, is now defined as paging space (/dev/hd6) under the new version.

However, this only applies to the Control Workstation. The SP nodes continue using /dev/hd7 as the primary dump device.

The following are the changes that occur when going from PSSP 1.2 to PSSP 2.1:

- Installation over DIX

  Support for installation over Ethernet type DIX has been added.

- Customization file

  firstboot.cust is now script.cust.

- Tuning customization

  A new tuning customization file called tuning.cust has been added.

- PSSP image

  bos.obj.ssp.325 is now bos.obj.ssp.41.

- mksysb image

  /usr/sys/inst.images/ssp is now /spdata/sys1/install/images.

  Under PSSP 1.2, the mksysb image could be a symbolic link. This is no longer the case. Network Installation Management (NIM) exports the file. The mksysb image must be a regular file in the */spdata/sys1/install/images* directory on the Control Workstation.

- ssp.installp

  ssp.installp is now pssp.installp.

- *spbootins*

  spbootins now has:

  – A diagnosis option, *-r diag*

  – New syntax for the *-h* option

- Remote Diagnosis Support has been added in PSSP 2.1.

2.2.14 PSSP Software Strategy

The PSSP updates and fixes are available on a regular basis from IBM as PTF sets. Some of the PSSP concepts and terminology are described as follows:

- PSSP PTFs are available as full replacement PTFs or delta replacement PTFs:

  – Full replacement PTFs

    Full replacement PTFs means that all of the associated codes in the affected component have been shipped in the PTF.

  – Delta replacement PTFs

    Delta replacement PTFs means that only the changed files from this PTF and prior delta replacement PTFs have been shipped in the PTF. These PTFs will have the last full replacement PTF as a prerequisite.

- PSSP PTF can be obtained through:

  FixDist

  You can obtain information about required PTFs from the Web at `http://www.rs6000.ibm.com/software/Pubs/sp_secure/status`. This page also contains a hyperlink to the US Fixdist server to download the required PTFs.

If you do not have direct access to the IBM network, then you may get these PTFs from the IBM Support Center by calling them and providing them with requirements, media type, delivery address, and urgency.

- Latest PSSP PTF

  This is always changing, so a good place to look at it is in the Web at http://www.rs6000.ibm.com/software/Pubs/sp_secure/status/, where you will find the latest PTF sets available, as well as a matrix describing the PSSP components affected by these PTFs.

2.2.15 Obtaining PSSP PTFs Using FixDist

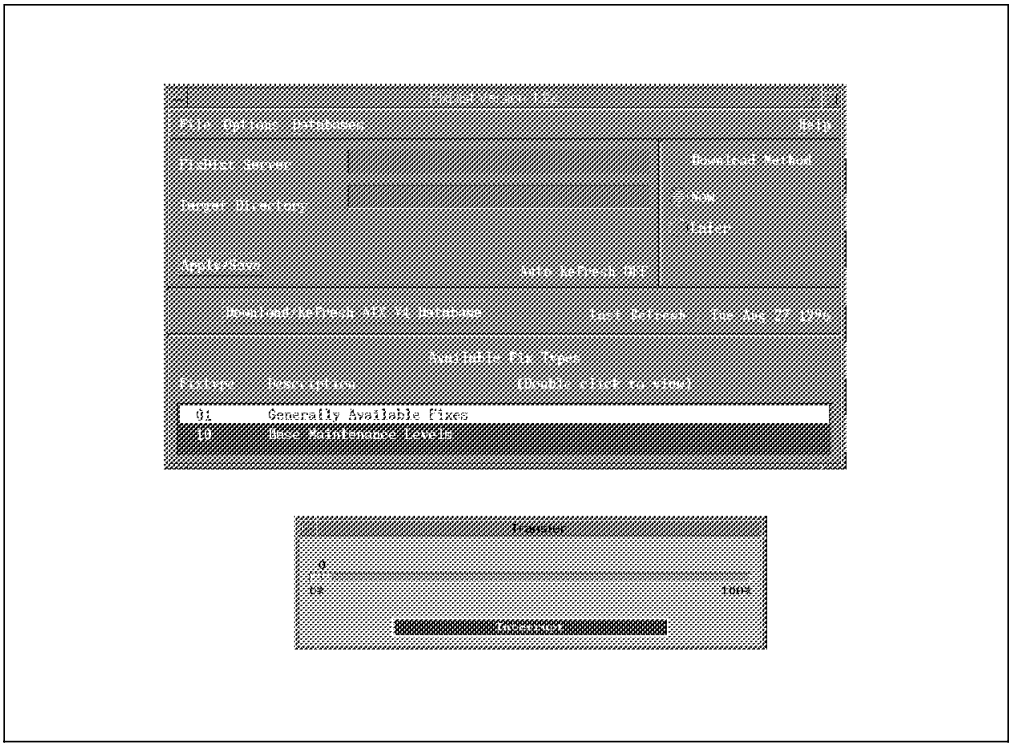


Figure 13. Obtaining PSSP PTFs Using FixDist

The easiest way to obtain AIX fixes and PSSP fixes is to use the tool FixDist. There are several servers available around the world that mirror the latest set of available PTFs:

- Canada rwww.aix.can.ibm.com 204.138.188.126
- Germany www.ibm.de 192.109.81.2
- Japan fixdist.yamato.ibm.co.jp 202.32.4.20
- Nordic ftp.nordic.ibm.com 193.12.214.80
- United Kingdom ftp.europe.ibm.com 193.129.186.2
- United States service.software.ibm.com 198.17.57.66

The FixDisk front end program shown in Figure 13 is also available on these servers.

2.2.16 Determining Software Service Level

It is important to find out what software service level you have because you may, for example, need it to report a problem to the IBM Support Center. The support staff needs this information to check various databases to ascertain if such a problem is already known on the software service level in question.

Specifying the current software service level while ordering PTFs dramatically reduces the PTF package size and installation time. There are at least two ways to find out your software service level:

- `lslpp` command

  The `lslpp` command has various flags to provide service level information, as the following example shows:

```
# lslpp -l | grep ssp
  ssp.authent           2.1.0.2  COMMITTED  SP Authentication Server
  ssp.basic            2.1.0.10  COMMITTED  SP System Support Package
  ssp.clients           2.1.0.5  COMMITTED  SP Authenticated Client
  ssp.css               2.1.0.8  COMMITTED  SP Communication Subsystem
  ssp.docs              2.1.0.4  COMMITTED  SP man and info files
  ssp.gui               2.1.0.6  COMMITTED  SP System Monitor Graphical
  ssp.jm                2.1.0.2  COMMITTED  SP Job Manager Package
  ssp.public            2.1.0.2  COMMITTED  Public Code Compressed
  ssp.sysctl            2.1.0.2  COMMITTED  SP Sysctl Package
  ssp.sysman            2.1.0.7  COMMITTED  Optional System Management
  ssp.top               2.1.0.2  COMMITTED  SP Communication Subsystem
  ssp.authent           2.1.0.0  COMMITTED  SP Authentication Server
  ssp.basic            2.1.0.10  COMMITTED  SP System Support Package
  ssp.clients           2.1.0.0  COMMITTED  SP Authenticated Client
  ssp.css               2.1.0.8  COMMITTED  SP Communication Subsystem
  ssp.jm                2.1.0.0  COMMITTED  SP Job Manager Package
  ssp.sysctl            2.1.0.0  COMMITTED  SP Sysctl Package
  ssp.sysman            2.1.0.7  COMMITTED  Optional System Management
  ssp.top               2.1.0.0  COMMITTED  SP Communication Subsystem
```

- Using PSSP Program Directory

  The structure is /usr/lpp/ssp/<*product*>/<*level*>, where *product* is an SSP product such as *ssp.basic*, and *level* is the level of service such as 2.1.0.10.

  You may traverse through the PSSP program directory for individual PSSP components and get to the *fixdata* file, as shown in the following example:

```
# cd /usr/lpp/ssp/ssp.basic/2.1.0.10
# ls -las
total 40
   4 drwxr-xr-x   3 root  system      512 May 14 15:43 .
   4 drwxr-xr-x   3 root  system      512 Apr 24 18:30 ..
   4 -r--r--r--   1 root  system     1902 Apr 23 17:13 README
   4 drwxr-xr-x   3 root  system      512 Apr 23 17:15 inst_root
   4 -rw-r--r--   1 root  system       15 Apr 23 17:13 productid
  12 -rw-r--r--   1 root  system     9630 Apr 23 17:13 ssp.basic.fixdata
# view ssp.basic.fixdata

fix:
        name = IX56314
        abstract = Vanderbilt PTF
        type = f
        filesets = "ssp.authent:2.1.0.2\n\
ssp.basic:2.1.0.10\n\
ssp.clients:2.1.0.5\n\
ssp.csd.cmi:2.1.0.1\n\
ssp.csd.hsd:2.1.0.2\n\
ssp.csd.vsd:2.1.0.4\n\
ssp.css:2.1.0.8\n\
ssp.docs:2.1.0.4\n\
ssp.gui:2.1.0.6\n\
ssp.jm:2.1.0.2\n\
ssp.public:2.1.0.2\n\
ssp.sysctl:2.1.0.2\n\
ssp.sysman:2.1.0.7\n\
ssp.top:2.1.0.2\n\
"
        symptom = ""

fix:
        name = IX54093
        abstract = Slot attrs to Frame class missing on install
"ssp.basic.fixdata" [Read only] 529 lines, 9630 characters
```

The lslpp command lists software products. It displays information about installed filesets or fileset updates. The FilesetName parameter is the name of a software product. The FixID (also known as PTF or program temporary fix ID) parameter specifies the identifier of an update to an AIX 3.2 formatted fileset.

When only the -l (lowercase L) flag is entered, the lslpp command displays the latest installed level of the fileset specified for the AIX 3.1 and 4.1 formatted filesets. The base level fileset is displayed for the AIX 3.2 formatted filesets.

When the -a flag is entered along with the -l flag, the lslpp command displays information about all installed filesets for the FilesetName specified.

The -I (uppercase i) flag combined with the -l (lowercase L) flag specifies that the output from the lslpp command should be limited to base level filesets.

The -d, -f, -h, -i, -l (lowercase L), -L, and -p flags request different types of output reports.

The -a, -c, -J, -O, and -q flags specify the amount and format of the information that is displayed in the report.

The default value for the FilesetName parameter is *all*, which displays information about all installed software products. Pattern matching characters, such as * (asterisk) and ? (question mark), are valid in the ProductName and FixID parameters. You do not have to enclose these characters in ′ ′ (single quotes). However, using single quotes prevents you from searching the contents of your present directory.

Much of the output from the `lslpp` command is understandable without an explanation. Other fields contain data that needs to be defined. The following table defines terms used in several of the output fields of the `lslpp` command:

Output field	Term	Definition
State of the fileset	APPLIED	The specified fileset is installed on the system. The APPLIED state means that the fileset can be rejected with the `installp` command, and the previous level of the fileset can be restored. This state is only valid for 4.1 fileset updates and 3.2 migrated filesets.
	APPLYING	An attempt was made to apply the specified fileset, but it did not complete successfully, and cleanup was not performed.
	BROKEN	The specified fileset or fileset update is broken and should be reinstalled before being used.
	COMMITTED	The specified fileset is installed on the system. The COMMITTED state means that a commitment has been made to this level of the software. A committed fileset update cannot be rejected, but a committed fileset base level and its updates (regardless of state) can be removed or deinstalled by the `installp` command.
	OBSOLETE	The specified fileset was installed with an earlier version of AIX (for example, 3.2), but it has been replaced by a repackaged (renamed) newer version. Some of the files that belonged to this fileset have been replaced by versions from the repackaged fileset.
	COMMITTING	An attempt was made to commit the specified fileset, but it did not complete successfully, and cleanup was not performed.
	REJECTING	An attempt was made to reject the specified fileset, but it did not complete successfully, and cleanup was not performed.
Action taken for the fileset	APPLY	An attempt was made to apply the specified fileset.
	CLEANUP	An attempt was made to perform cleanup for the specified fileset.
	COMMIT	An attempt was made to commit the specified fileset.
	REJECT	An attempt was made to reject the specified fileset.

Output field	Term	Definition
Status history results of install actions	BROKEN	The fileset was left in a broken state after the specified action.
	CANCELED	The specified action was canceled before it completed.
	COMPLETE	The commitment of the fileset has completed successfully.
	NONE	This fileset update has not been installed but a superseding update has (applicable to AIX 3.2 formatted fileset updates only).

Table 3. Output Terms of the lslpp Command

Examples:

1. To list the installation state for the most recent level of installed filesets for all of the bos.rte filesets, enter:

   ```
   lslpp -l "bos.rte.*"
   ```

2. To list the installation state for the base level and updates for the fileset bos.rte.filesystem, enter:

   ```
   lslpp -La bos.rte.filesystem
   ```

3. To list the installation history information of all the filesets in the bos.net software package, enter:

   ```
   lslpp -ha 'bos.net.*'
   ```

4. To list the names of all the files of the bos.rte.lvm fileset, enter:

   ```
   lslpp -f bos.rte.lvm
   ```

2.2.17 Working with PSSP PTFs

PTFs are an essential part of PSSP installation. By installing correct levels of PSSP PTFs, you ensure that the PSSP software enhancements and fixes for known problems are incorporated in the PSSP code on your Control Workstation and in your nodes.

When dealing with PTFs, you should be aware of the following:

- Check smit.log

  Always use `smit` when you are dealing with PTFs.

  The `smit` command will give you the status of completion in the smit.log file. After installation has completed (or FAILED), check the smit.log thoroughly and identify any instances of FAILED messages. Determine which components, if any, have FAILED to install. The `smit` command provides information as to why the install or upgrade has failed, and what you need to do to fix the problem. Take actions required to rectify the problem. Use of the `smit` command can be very helpful in determining why the problem occurred, which part is affected, and how to fix the problem.

- PTF prerequisites, corequisites, and if-requisites

  Always ensure that the PTF you are installing has all the prerequisites, corequisites and if-requisites. If you are missing any prerequisites, then PTF install will fail. *smit.log* will provide information as to what was missing and what to do next.

- README file

  Always read the README file under directory /usr/lpp/ssp for information about various PSSP components, including product information, installation information, restrictions, advisories, important APAR information, and other allied information. This can be very useful information in resolving problems even before they arise. Here is the listing of the */usr/lpp/ssp/README* directory:

```
# cd /usr/lpp/ssp/README
# ls -l
total 152
-rw-r--r--   1 bin   bin    4110 Apr  9 06:45 ssp.authent.README
-rw-r--r--   1 bin   bin    3454 Apr 12 14:31 ssp.authent.README.IX56801
-rw-r--r--   1 bin   bin    5372 Apr  9 07:05 ssp.basic.README
-rw-r--r--   1 bin   bin     127 Apr  9 09:39 ssp.basic.README.IX53006
-rw-r--r--   1 bin   bin     247 Apr  9 09:39 ssp.basic.README.IX54269
-rw-r--r--   1 bin   bin    2945 Apr  9 07:05 ssp.basic.README.IX56314
-rw-r--r--   1 bin   bin    3608 Apr  9 06:49 ssp.clients.README
-r--r--r--   1 bin   bin    1524 Apr  9 07:00 ssp.css.README
-r--r--r--   1 bin   bin     402 Apr  9 07:05 ssp.css.README.IX56314
-rw-r--r--   1 bin   bin    5907 Apr  9 07:00 ssp.docs.README
-r--r--r--   1 bin   bin    1863 Apr  9 06:45 ssp.gui.README
-rw-r--r--   1 bin   bin    1386 Apr  9 06:44 ssp.jm.README
-rw-r--r--   1 bin   bin    3403 Apr  9 06:44 ssp.public.README
-rw-r--r--   1 bin   bin    2186 Apr  9 09:34 ssp.sysctl.README
-rw-r--r--   1 bin   bin    1487 Apr  9 06:44 ssp.sysman.README
-rw-r--r--   1 bin   bin     116 Apr  9 09:39 ssp.top.README.IX53362
```

  README files also exist for some products under /usr/lpp/ssp/*product/level*, particularly after you have already installed some PTFs for that component. For example, here is the listing of the */usr/lpp/ssp/ssp.basic/2.1.0.10* directory after installing PTFset11:

```
# cd /usr/lpp/ssp/ssp.basic/2.1.0.10
# ls -las
 4 drwxr-xr-x   3 root     system     512 Apr 23 17:16 .
 4 drwxr-xr-x   3 root     system     512 Apr 24 18:30 ..
 4 -r--r--r--   1 root     system    1902 Apr 23 17:13 README
 4 drwxr-xr-x   3 root     system     512 Apr 23 17:15 inst_root
 4 -rw-r--r--   1 root     system      15 Apr 23 17:13 productid
12 -rw-r--r--   1 root     system    9630 Apr 23 17:13 ssp.basic.fixdata
```

- Stop processes (if required)

  Always read the PSSP component's README file under */usr/lpp/ssp/README* before installing any PTF that affects a component. If, for example, you do not read *ssp.authent.README.IX56801* under the README directory and install PTFset11, which also updates *ssp.authent*, then the installation will fail as if the Kerberos daemon were still running.

  If you try to install PTF for *ssp.authent* after the *setup_authent* step, then the PTF installation will fail. You must not have kerberos, kadmind, or hardmon running when installing the PTF for *ssp.authent*. If you do, smit.log will show the following:

```
ERROR:
  kadmind is running.
  kerberos is running.
  Because of this, the PTF installation cannot go on.
  Stop daemons kerberos, kadmind and hardmon.
  Install the PTF.
  Then restart those three daemons.

  For further information, see the ssp.authent.README.IX56801 that
  accompanies this PTF.
update:  Failed while executing the ssp.authent.pre_u script.

0503-464 installp:  The installation has FAILED for the "usr" part
        of the following filesets:
        ssp.authent 2.1.0.2

installp:  Cleaning up software for:
        ssp.authent 2.1.0.2

Finished processing all filesets.  (Total time:  11 mins 42 secs).
```

Execute the following steps before re-installing the PTF:

```
# chitab "kadm:2:off:/usr/lpp/ssp/kerberos/etc/kadmind -n"
# chitab "kerb:2:off:/usr/lpp/ssp/kerberos/etc/kerberos"
# telinit 2
# stopsrc -s hardmon
# smit install_selectable_all
```

If the installation fails, check smit.log and search for FAIL. Take corrective action as required.

After the successful installation of the PTF, you have to perform the following steps to change inittab file and start the kadmind, kerberos, and hardmon daemons.

```
# chitab "kadm:2:respawn:/usr/lpp/ssp/kerberos/etc/kadmind
# chitab "kerb:2:respawn:/usr/lpp/ssp/kerberos/etc/kerberos
# telinit 2
# startsrc -s hardmon
```

- Loss of sysman configuration files after installing PTFSet11

  PTFset11 for RS/6000 SP Version 2 Release 1 is a full replacement PTF set and will therefore replace some configuration files that you have on your system. To retain the contents of these files, you need to save and then restore them after the installation of PTF Set 11. The following is a list of the *ssp.sysman 2.1.0.7* files that will be replaced:

  ```
  /usr/lpp/ssp/filec/file.collections
  /usr/lpp/ssp/filec/sup/sup.admin/list
  /usr/lpp/ssp/filec/sup/sup.admin/lock
  /usr/lpp/ssp/filec/sup/sup.admin/prefix
  /usr/lpp/ssp/filec/sup/sup.admin/refuse
  /usr/lpp/ssp/filec/sup/sup.admin/supperlock
  /usr/lpp/ssp/filec/sup/user.admin/list
  /usr/lpp/ssp/filec/sup/user.admin/lock
  /usr/lpp/ssp/filec/sup/user.admin/prefix
  /usr/lpp/ssp/filec/sup/user.admin/refuse
  /usr/lpp/ssp/filec/sup/user.admin/supperlock
  /usr/lpp/ssp/filec/sup/node.root/list
  /usr/lpp/ssp/filec/sup/node.root/lock
  ```

```
/usr/lpp/ssp/filec/sup/node.root/prefix
/usr/lpp/ssp/filec/sup/node.root/refuse
/usr/lpp/ssp/filec/sup/node.root/supperlock
/usr/lpp/ssp/filec/sup/power_system/list
/usr/lpp/ssp/filec/sup/power_system/lock
/usr/lpp/ssp/filec/sup/power_system/prefix
/usr/lpp/ssp/filec/sup/power_system/refuse
/usr/lpp/ssp/filec/sup/power_system/supperlock
```

You should never run base level PSSP software by itself. At minimum, you should be on PTFset4 or PTFset8. PTFset11 is the latest full replacement set that can be installed on base PSSP 2.1 software level. Various defects have been fixed and feature enhancements have been made on the PSSP base level of software. PTFs are available through Support Organizations, or online through *fixdist*.

Table 4 lists various components and their sizes of PTFset11 available at the time this document was written.

Component	Size
ppe.poe.2.1.0.8	2063360
ppe.vt.2.1.0.4	1428480
ssp.authent.2.1.0.2	380928
ssp.basic.2.1.0.10	17014784
ssp.clients.2.1.0.5	1904640
ssp.csd.cmi.2.1.0.1	158720
ssp.csd.hsd.2.1.0.2	190464
ssp.csd.vsd.2.1.0.4	317440
ssp.css.2.1.0.8	6221824
ssp.docs.2.1.0.4	10412032
ssp.gui.2.1.0.6	1777664
ssp.jm.2.1.0.2	1333248
ssp.public.2.1.0.2	12348416
ssp.sysctl.2.1.0.2	539648
ssp.sysman.2.1.0.7	1111040
ssp.top.2.1.0.2	984064

Table 4. Components and Sizes of the PSSP PTFset11

2.3 Authentication Services Diagnostics

To initialize your primary authentication server on the RS/6000 SP Control Workstation, you need to run the setup_authent command from the command line on your Control Workstation. The Kerberos administrator needs to define a kerberos master key, principal name, instance, password for your principal and instance, and then again password before kerberos initialization for your principal.instance can start. Step-by-step instructions of your interaction with setup_authent is explained in the SP Installation Guide. Details of what setup_authent does is explained in Chapter 3, “Kerberos” on page 71 under 3.4.1, “setup_authent” on page 74.

Path Problems If running `setup_authent` gives the error `setup_authent not found`, then check your path for `setup_authent`. You can use `which setup_authent` command to get the correct path to `setup_authent`. Include correct kerberos paths in your .profile file or use full path.

Here are the paths required by kerberos that you should put in your .profile file:

```
/usr/lpp/ssp/rcmd/bin
/usr/lpp/ssp/kerberos/bin
/usr/lpp/ssp/kerberos/etc
/usr/lpp/ssp/bin
```

Kerberos Problems For tips to solve various kerberos problems, refer to 3.8, "Solving Problems" on page 82 in Chapter 3, "Kerberos" on page 71. You may also take a look at the setup_authent script flow chart in A.1, "The setup_authent Script" on page 229.

2.4 install_cw Diagnostics

Before you proceed further, ensure that you are authenticated as a valid user with a valid ticket. If required, run the `kinit` command to get a valid ticket from RS/6000 SP authentication services specifying the administrative principal name that was used when authentication was set up.

The full path of this script is /usr/lpp/ssp/bin/install_cw.

install_cw runs on the Control Workstation and does the following:

- Configures the Control Workstation
- Executes */usr/lpp/ssp/inst_root/ssp.basic.post_i*

  After PTF 12, this script is not called directly.

  There was a dependency on ssp.basic.post_i after installation which caused several problems. It was most easily seen when a node had been restored from a mksysb in which the variable RM_INST_ROOT is set to yes in the bosinst.data on the system from which the mksysb was taken. This caused `inurid -r` to be run when the system was restored.

  `pssp_script` calls `ssp.basic.post_i` during install and fails to customize the node properly after installation.

  The pssp_script has been changed to call post_process and post_sysctl. The post_process and post_sysctl files are equivalent to the post_i ssp.basic.post_i and ssp.sysctl.post_i files, respectively, and they are located in the /usr/lpp/ssp/install/bin directory.
- Installs *smit* panels
- Configures SDR
- Updates */etc/services* and */etc/inittab*
- Starts SP daemons
- Creates */spdata/sys1/spmon/hmacls* file with the administrator's ACL for hardmon
- Configures default system partition

- For a complete flow chart of what install_cw does, refer to Appendix A.2, “The install_cw Script” on page 236.

Figure 14 shows a sample output of running the install_cw script.

```
13 entries added.
0 entries deleted.
0 entries updated.
0513-071 The sdr.sp2cw0 Subsystem has been added.
making SRC object "hb.sp2cw0" at system level PSSP-2.1
0513-071 The hb.sp2cw0 Subsystem has been added.
making SRC object "hr.sp2cw0"
0513-071 The hr.sp2cw0 Subsystem has been added.
0513-085 The hardmon Subsystem is not on file.
0513-084 There were no records that matched your request.
0513-071 The hardmon Subsystem has been added.
13 entries added.
0 entries deleted.
1 entries updated.
0513-085 The splogd Subsystem is not on file.
0513-084 There were no records that matched your request.
0513-071 The splogd Subsystem has been added.
0513-059 The splogd Subsystem has been started. Subsystem PID is 15102.
0513-059 The sysctld Subsystem has been started. Subsystem PID is 16390.
```

Figure 14. Output of Running install_cw Script

Figure 15 shows what gets added to the /etc/inittab file from running the install_cw script.

```
sdrd:2:once:/usr/bin/startsrc -g sdr
sp:2:wait:/etc/rc.sp > /dev/console 2>&1
hardmon:2:once:/usr/bin/startsrc -s hardmon
hr:2:once:/usr/bin/startsrc -g hr
hb:2:once:/usr/bin/startsrc -g hb >/dev/null 2>/dev/console
splogd:2:once:/usr/bin/startsrc -s splogd
```

Figure 15. Portion of /etc/inittab That Gets Added from Running install_cw

Figure 16 shows what gets added to /etc/services from running the install_cw script.

```
hardmon              8435/tcp
sdr                  5712/tcp
heartbeat     4893/udp
```

Figure 16. Portion of /etc/services That Gets Added from Running install_cw

2.4.1 Problems Solving

If you have problems running the install_cw script, check following:

- /usr/lpp/ssp/bin should be exported in your PATH.
- You should be logged on as a valid authenticated user of the system management commands.

- You should have a valid ticket from the RS/6000 SP authentication services. Use the `klist` command to check authentication and `kinit` command to request authentication (if required).
- If the install_cw script does not complete successfully, then refer to Appendix A.2, “The install_cw Script” on page 236 to see a detailed flowchart of the working of this script and identify where the script has failed. *Diagnosis and Message Guide*, GC23-3899 could provide useful pointers based on the error message that you received. For example, if the script has failed because you have not yet completed the kerberos configuration, then you need to complete the authentication part first.

2.5 Problems with spmon

SP System Monitor refers to four main components:

- Hardware Monitor

  The hardware monitor consists of a set of commands and a daemon to monitor and control the SP hardware platform

- SP System Monitor GUI

  The SP system monitor graphical user interface (GUI) to the hardware monitor provides a point-and-click method to view the system status and interact with the frame, node, and switch controls.

- Command Line Interface

  The `spmon`, `hmmon`, `hmcmds`, `s1term`, and `nodecond` commands with their flags and parameters, provide a way to monitor and control the SP system.

- Logging daemon

  The splogd logging daemon logs SP hardware state changes, reports SP hardware errors, and provides a state change alert.

Figure 17 on page 32 shows a diagram of a System Monitor, where lines represent relationships of its different components.

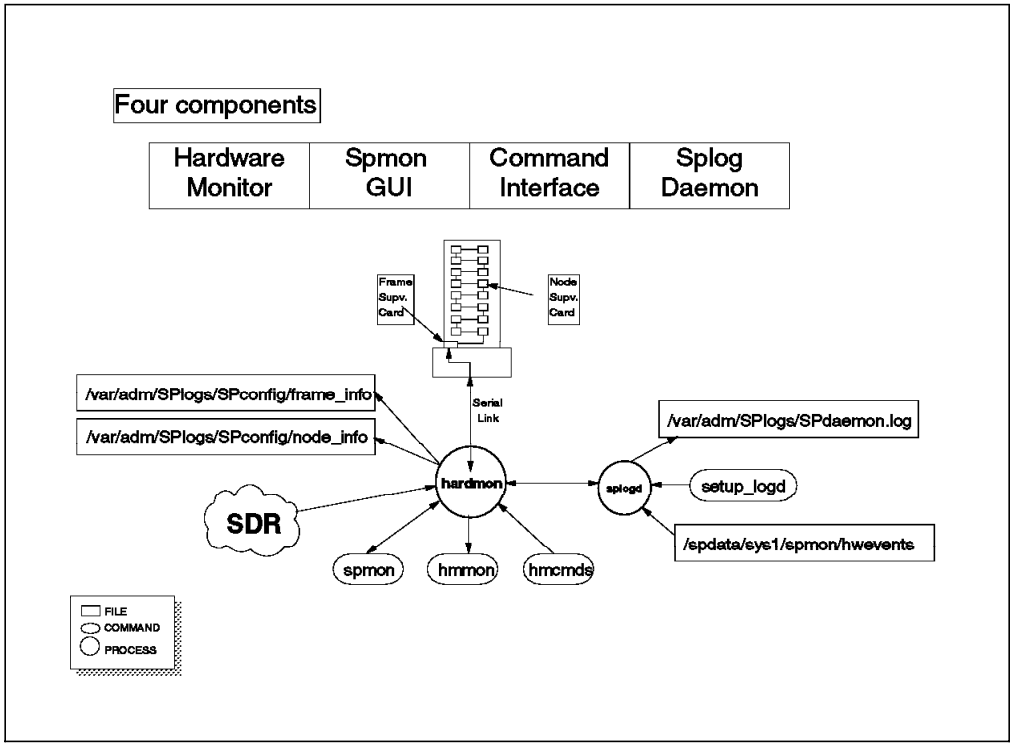


Figure 17. Snapshot of System Monitor

The hardware monitor consists of a daemon, named hardmon, and a set of client commands. The hardmon daemon executes on the Control Workstation and, using the RS-232 lines to each frame, polls the frames for the state of the hardware within the frame. The daemon also sends hardware-level commands to each frame to change the state of the hardware within the frame in some way.

The client commands `spmon` and `hmmon` interact with hardmon to obtain the current or changing state of the hardware.

The client commands `spmon` and `hmcmds` also interact with hardmon to change the state and control the hardware. The SP system monitor GUI issues the `spmon` command internally to control and monitor the hardware.

The splogd logging daemon is also a client of the hardware monitor for logging purposes.

2.5.1 Problems with System Monitor Commands

When you try to run `spmon -g` on the Control Workstation and the system reports that `spmon: not found.`, then verify installation, as follows:

- Make sure that the *ssp.basic* option of the *pssp installp* image was installed. The *ssp.basic* option provides the system monitoring function.

  Run verification tests using `smit` or the command line to ensure installation is complete. Use `smit` as follows:

```
# smit smoni_verify
   or enter the following on the command line:
# /usr/lpp/ssp/bin/spmon_itest
```

```
sysmon_itest: Start sysmon installation verification test
sysmon_itest: Verification Succeeded
```

This result indicates that the *ssp.basic* option is installed.

- If the verification in the previous step succeeds, then check your PATH environment variable. You can achieve this by using echo $PATH.
  - Add /usr/lpp/ssp/bin to your PATH or use the full path name with the command.

2.5.2 Problems Accessing Pulldown Menus of spmon

On the Control Workstation, if you run `spmon -g` in order to get the graphic interface of `spmon`, and the numlock key is on, then you will not be able to access any menu in `spmon` by clicking on Files or SP with the mouse. If numlock is off, `spmon` works without any problem. When you experience this problem with `spmon` with numlock on, you can still access the mwm root menu and aixterms window menus. This problem is reported under IX50314. This problem was tested on the ssp.gui.2.1.0.6 level.

2.5.3 System Monitor Displays Blanks Instead of the Expected Information

If the system monitor panels are displaying blanks instead of the expected information after installation, or if the ControllerResponds indicator on the system monitor frame environment panel is red, or if the /var/adm/SPlogs/SPdaemon.log reports that the Frame Controller is not responding, then do the following:

- Verify PSSP Installation.

  Run verification tests to ensure that installation is complete. You can achieve this using `smit` or from the command line. Use `smit` as follows:

  ```
  # smit sman_verify
  ```

  or, from the command line, enter the following:

  ```
  # /usr/lpp/ssp/bin/SYSMAN_test
  ```

```
HOST: sp2cw0
------------
SYSMAN_test: Verification succeeded

SYSMAN_test: Executing test on all active nodes

HOST: sp2n01
------------
SYSMAN_test: Verification succeeded

HOST: sp2n02
------------
SYSMAN_test: Verification succeeded

HOST: sp2n03
------------
SYSMAN_test: Verification succeeded
  .             .
  .             .
  .             .
HOST: sp2n13
------------
SYSMAN_test: Verification succeeded

HOST: sp2n14
------------
SYSMAN_test: Verification succeeded
SYSMAN_test: The number of nodes that were not tested is 4
```

This output confirms that your installation is complete.

- Verify the RS-232 connection.

  Ensure that the RS-232 line is connected to the correct frame and the correct serial port on the Control Workstation. Remember that the hardmon daemon on the Control Workstation requires an RS-232 connection to each frame to poll the frame for the state of hardware within the frame. Without the RS-232 line, hardmon has no way to check the state of hardware in the frame.

  Again, you have the choice of using smit or the command line. Use smit as follows:

  ```
  # smit smonc_verify
  ```

  or, from the command line, enter the following:

  ```
  # /usr/lpp/ssp/bin/spmon_ctest
  ```

  ```
  sysmon_ctest: Start sysmon configuration verification test
  sysmon_ctest: Verification Succeeded
  ```

  This output confirms that your RS-232 connection is fine.

- Verify Serial Port speed.

  Check the baud rate of the serial port. The System Monitor configures this for you when it starts. Use the stty command as follows to redirect input from the correct serial port:

  ```
  # stty </dev/tty0
  speed 19200 baud; -parity clocal
     .                 .
     .                 .
  -isig -icanon -echo -echoe -echok
  ```

  In this example, *speed 19200* is correct.

- Check frame configuration.

  If the frame information is missing when you use any one of the following commands, the SDR does not know about the frame. Accordingly, spmon -g will not be able to start the spmon gui.

  You can verify this by using either smit or the command line. Use smit as follows:

  ```
  # smit list_frame
  ```

  or, from the command line, enter the following:

  ```
  # SDRGetObjects Frame
  frame_number tty          frame_type   MACN         backup_MACN
             1 /dev/tty0    switch       sp2cw0       ""
  #
  ```

  or

  ```
  #
  # /usr/lpp/ssp/bin/splstdata -f
                    List Frame Database Information

  frame#              tty       frame_type
  ------------------------------------
       1        /dev/tty0          switch
  #
  ```

  The output illustrated confirms that the frame information is available.

If you find out that the frame information is missing, you first need to add frame information, and then reinitialize the SDR. This can be done through the smit fastpath smit frame_dialog.

- Check the log for messages.

  Check the /var/adm/SPlogs/SPdaemon.log on the control workstation and take action to address any messages. Look for resource name *sphwlog* for SP hardware problems.

```
May  8 20:03:09 sp2cw0 hardmon[13970]:
LPP=PSSP,Fn=hardmon.c,SID=1.26,L#=260,
 hardmon: 0026-874 Unable to determine whether
 the Control Workstation is active, inactive, or non-HACWS.
```

  In the /var/adm/SPlogs/SPdaemon.log file on the Control Workstation, there was no entry with *sphwlog*. The preceding example is an entry that was found and is shown here just as an example entry in the log file. The same messages are also logged under *errpt*.

  – Check /var/adm/SPlogs/SPdaemon.log for messages containing the word sphwlog.

  – Perform the following command to check the AIX error log:

```
# errpt -aN sphwlog
LABEL:          SPMON_INFO101_TR
IDENTIFIER:     3E6F3CE7

Date/Time:       Tue May  7 15:15:07
Sequence Number: 965
Machine Id:      000081007000
Node Id:         sp2cw0
Class:           H
Type:            UNKN
Resource Name:   sphwlog
Resource Class:  NONE
Resource Type:   NONE
Location:        NONE

Description
ELECTRICAL POWER RESUMED

Probable Causes
POWER SUBSYSTEM

User Causes
POWER OFF
        Recommended Actions
        NONE

Detail Data
DETECTING MODULE
LPP=PSSP,Fn=splogd.c,SID=1.16.1.7,L#=909,
DIAGNOSTIC EXPLANATION
Information; Node 1:8; powerLED; Power is on.
#
```

    This log provides full details of all SP hardware errors.

- If you cannot figure out the cause of the problem, call the IBM Support Center.

2.5.4 Problems with System Monitor GUI

- If the spmon command fails to start the System Monitor GUI, you need to check the following:
  - Verify authorization:
    - Run the klist command to verify that the kerberos tickets are not expired.
    - Reissue the kinit command if required.
    - Check /spdata/sys1/spmon/hmacls to ensure that the kerberos principal name and instance for your ID is in this hardware Monitor ACL file:

```
sp2cw0 root.admin a
sp2cw0 hardmon.sp2cw0 a
1 root.admin vsm
1 hardmon.sp2cw0 vsm
```

  - Ensure you have exported DISPLAY:

```
# echo $DISPLAY
9.12.0.3:0
```

    In this example, DISPLAY is exported correctly to the local display on the PC with an IP address 9.12.0.3.

- If an open session failure message window is displayed after opening the System Monitor GUI, you need to check following:
  - Verify authorization using klist, request authorization using kinit, and check /spdata/sys1/spmon/hmacls.
  - Check hardware monitor daemon (hardmon):
    - Issue the hardware monitor command using an ID that has monitor authority and a valid ticket-granting ticket from kinit.

      Following is an example command to check frame 1, node 1:

      ```
      # hmmon -Q 1:1
      ```

      If this command does not work, check the hardmon log /var/adm/SPlogs/spmon/hmlogfile, where nnn is the Julian date of when the file was created.
    - Check /var/adm/SPLogs/spmon/hmlogfile.nnn

- If the System Monitor GUI hangs, it could be because of performance reasons or authorization problems. Check the following:
  - Check performance:
    - Check for adequate paging space.
    - Check overall CPU utilization.
    - Check CPU utilization of hardmon and splogd:

      ```
      # ps vgc | grep hardmon
      # ps vgc | grep splogd
      ```

      If the CPU utilization rate is very high and cannot be attributed to the hardmon or logging daemon, then look for other processes which are consuming the CPU resources. Contact the IBM Support Center if you cannot resolve the problem.

- Verify authorization through klist.

2.5.5 Logging Problems

- If the logfile is no longer being updated:

  Check the logging daemon:

  - Check that splogd is running.

    The following example shows how you can check that splogd is running using either the lssrc -s or the ps -eaf command:

```
# lssrc -a | grep splogd
 splogd                           15378   active
#
# ps -eaf | grep splogd
    root 15378  5854 0   May 08  -  0:00 /usr/lpp/ssp/bin/splogd
-f /spdata/sys1/spmon/hwevents
    root 18408 15248 0 11:22:43  pts/0 0:00 grep splogd
```

  - Check /etc/syslog.conf for daemon.notice entry

```
daemon.notice           /var/adm/SPlogs/SPdaemon.log
```

  - Check that the /var/adm/SPlogs/SPdaemon.log exists with permissions rw-r--r--.
  - Check that error record templates for SPMON exist:

    # errpt -t | grep SPMON

- If the logging daemon dies:
  - Check for a core dump. If available, run crash to identify the reason for the core dump. If required, send the core dump and /unix file to the IBM Support Center for investigation.
- If the hardware monitor daemon (hardmon) dies:
  - Check for a core dump. If available, run crash to identify the reason for the core dump. If required, send the core dump and /unix file to the IBM Support Center for investigation.
  - Check /var/adm/SPlogs/spmon/hmlogfile.nnn.

2.5.6 Example — Some Nodes Are Missing

This is an example of how to investigate if spmon does not show a node. In this example, assume node 8 is not showing up.

1. Run splstdata -n. This proves whether or not SDR has information about this node.

```
# splstdata -n
                    List Node Configuration Information

node# frame# slot# slots  initial_hostname reliable_hostname default_rout
------------------------------------------------------------------------------
    1      1     1     1  sp2n01             sp2n01            9.12.20.99
    2      1     2     1  sp2n02             sp2n02            9.12.20.99
    3      1     3     1  sp2n03             sp2n03            9.12.20.99
    4      1     4     1  sp2n04             sp2n04            9.12.20.99
    5      1     5     1  sp2n05             sp2n05            9.12.20.99
    6      1     6     1  sp2n06             sp2n06            9.12.20.99
    7      1     7     1  sp2n07             sp2n07            9.12.20.99
    8      1     8     1  sp2n08             sp2n08            9.12.20.99
    9      1     9     1  sp2n09             sp2n09            9.12.20.99
   10      1    10     1  sp2n10             sp2n10            9.12.20.99
   11      1    11     1  sp2n11             sp2n11            9.12.20.99
   12      1    12     1  sp2n12             sp2n12            9.12.20.99
   13      1    13     1  sp2n13             sp2n13            9.12.20.99
   14      1    14     1  sp2n14             sp2n14            9.12.20.99
   15      1    15     1  sp2n15             sp2n15            9.12.20.99
   16      1    16     1  sp2n16             sp2n16            9.12.20.99
```

In this case, SDR has information about node 8.

2. Check /var/adm/SPlogs/spmon and view hmlogfile.nnn.

   The following is a listing of the relevant portion of the /var/adm/SPlogs/SPconfig/node_info file:

   ```
   /SP/frame/frame1/node7/type/value/97
   /SP/frame/frame1/node9/type/value/97
   ```

   node_info is missing node 8.

3. Try `s1term -w 1 8`. The following message is received:

   ```
   s1term: 0026-645 The S1 port in frame 1 slot 8 cannot be accessed.
   It either does not exist or you do not have S1 permission.
   ```

4. Check the RS-232 cable from node 8 to the switch:

   If a cable is loose, securing it will cause spmon to show node 8.

2.5.7 Example — spmon Will Not Start

The actions to be taken will depend on the error message displayed by the spmon GUI when it fails.

1. Check that the hardmon daemon is running.

Figure 18. The hardmon Daemon Is Not Running

`ps -eaf | grep hardmon` shows that it is running, but another `ps -eaf` shows it is continually respawning. `fuser /dev/tty0` reconfirms this.

2. Check that the sdr daemon is running by using

   `lssrc -g sdr` or `ps -aef | grep sdr`

Figure 19. The sdr Daemon Is Not Running

If the sdr daemon is not running, enter the following:

```
startsrc -g sdr
```

If the problem persists, move on to the next check.

3. Check the /spdata/sys1/spmon/hmacls file.

Figure 20. The hmacls File

In this example, /spdata/sys1/spmon/hmacls has incomplete entries. The following two lines are missing:

```
1 root.admin vsm
1 hardmon.sp2cw0 vsm
```

Even if these lines are added by hand, there may still be a problem starting spmon. Check for and remove any blank lines.

4. Kill hardmon to respawn it with new information from the /spdata/sys1/spmon/hmacls file.

After killing hardmon, spmon -g will start okay.

2.6 Problems with Some PSSP Scripts

2.6.1 setup_authent Script

If you have problems running setup_authent script, check the following:

- PATH

  /usr/lpp/ssp/bin should be exported in your PATH.

- Name resolution

For all configured IP interfaces, the name resolution has to be set up correctly by using the /etc/hosts file or DNS. This also includes interfaces other than the SP internal Ethernet.

- Configuration files

  In case you are using an existing Kerberos server, make sure the configuration files /etc/krb.conf and /etc/krb.realms are set up correctly. Remove these files in case you employ the Control Workstation as a Kerberos server. The configuration files will be created by setup_authent according to your Control Workstations hostname, domain name, or both.

There might be cases in which you want to run this script in whole a second time. You can do this, but be aware that performing this action will destroy the Kerberos database. *Never do this on a running SP system!*

1. Remove the Kerberos configuration files and the Kerberos server keytab file.

```
# rm -i /etc/krb*
rm: Remove /etc/krb-srvtab? y
rm: Remove /etc/krb.conf? y
rm: Remove /etc/krb.realms? y
#
```

2. Remove the Kerberos master key file and the ticket cache file.

```
# rm -i /.k /tmp/tkt0
rm: Remove /.k? y
rm: Remove /tmp/tkt0? y
#
```

3. Remove the Kerberos user database files.

```
# rm -i /var/kerberos/database/*
rm: Remove /var/kerberos/database/admin_acl.add? y
rm: Remove /var/kerberos/database/admin_acl.get? y
rm: Remove /var/kerberos/database/admin_acl.mod? y
rm: Remove /var/kerberos/database/principal.dir? y
rm: Remove /var/kerberos/database/principal.ok? y
rm: Remove /var/kerberos/database/principal.pag? y
rm: Remove /var/kerberos/database/slavesave? y
#
```

4. Correct any mistakes you have made, for example incomplete name resolution, and rerun the script:

   ```
   # setup_authent
   ```

5. Make sure you now can login as root.admin@<REALM> and the service keys for rcmd and hardmon have been created.

```
# kinit root.admin
Kerberos Initialization for "root.admin"
Password: <password>
# klist
Ticket file:    /tmp/tkt0
Principal:      root.admin@SPCNTL

  Issued           Expires          Principal
Jul 17 17:02:51  Aug 16 17:02:51  krbtgt.SPCNTL@SPCNTL
# klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
rcmd            spcntl          SPCNTL          1
hardmon         spcntl          SPCNTL          1
#
```

6. If you have problems running the setup_authent script, you may use the flow chart in A.1, “The setup_authent Script” on page 229 and see where this script is failing. What actions you follow will depend on the error message you are getting from running the script. However, it is very uncommon for setup_authent to fail when the entire database has been deleted following the previous steps.

2.6.2 install_cw Script

If you have problems running the install_cw script, check the following:

- PATH

  /usr/lpp/ssp/bin should be exported in your PATH.

- Authentication

  You should be logged in as a valid authenticated user of the system management commands.

- Valid ticket

  You should have a valid ticket from the RS/6000 SP authentication services. Use the klist command to check authentication, and the kinit command to request authentication (if required).

- install_cw Flow Chart

  If the install_cw script does not complete successfully, refer to the detailed flow chart of the working of this script in A.2, “The install_cw Script” on page 236, and identify where the script has failed. *SP Diagnosis and Messages Guide*, GC23-3899 could provide useful pointers based on the error message that you received. For example, if the script has failed because you have not yet completed *kerberos* configuration, then you need to complete the authentication part first.

Figure 21 on page 43 shows what gets added to the /etc/inittab file by running the install_cw script.

```
sdrd:2:once:/usr/bin/startsrc -g sdr
sp:2:wait:/etc/rc.sp > /dev/console 2>&1
hardmon:2:once:/usr/bin/startsrc -s hardmon
hr:2:once:/usr/bin/startsrc -g hr
hb:2:once:/usr/bin/startsrc -g hb >/dev/null 2>/dev/console
splogd:2:once:/usr/bin/startsrc -s splogd
```

Figure 21. /etc/inittab File after Running install_cw Script

The next figure shows what gets added to the /etc/services by running the install_cw script.

```
hardmon               8435/tcp
sdr                   5712/tcp
heartbeat     4893/udp
```

Figure 22. /etc/services File after Running install_cw Script

2.7 Restore Control Workstation mksysb

There might be situations in which you have to rerun the script install_cw. For example, perhaps you have restored a mksysb of the Control Workstation to recover from the loss of a disk. Each node of the SP and the CWS has a *node_number* entry in its ODM (CuAt). This entry gets lost when you restore the mksysb image on the CWS. There are two different methods you can use to recreate this entry.

1. Run install_cw. This will create the proper node_number entry for the Control Workstation in the ODM. This script also performs other actions that do not affect the configuration of the CWS and which do not need to be run to recreate the entry in the ODM. For instance, the splogd and hardmon subsystems are deleted and recreated.
2. The other method is to create the entry in the ODM:
   a. Make sure that there is no old entry in the ODM:

      ```
      # odmdelete -o CuAt -q "name=sp and attribute=node_number"
      ```

   b. Copy the template file for the node_number entry to some temporary disk space:

      ```
      # cp /usr/lpp/ssp/install/config/cuat.sp /tmp/cuat.sp
      ```

   c. Edit this file and replace the line containing the attribute *value = "NnN"* with *value = "0"*, reflecting that the node number of the CWS is "0."

   d. Insert this entry into the ODM:

      ```
      # ODMDIR=/etc/objrepos odmadd /tmp/cuat.sp
      ```

   e. Verify that the entry has been created, and remove the temporarily created file:

```
# odmget -q name=sp CuAt
CuAt:
        name = "sp"
        attribute = "node_number"
        value = "0"
        type = "R"
        generic = "DU"
        rep = "s"
        nls_index = 24

# rm /tmp/cuat.sp
```

2.8 NIM Problems

Network Installation Management (NIM) enables you to centrally manage the installation of AIX Version 4.1 Base Operating System (BOS) and optional software on machines with a networking environment.

As part of PSSP 2.1, NIM was introduced as the network installation support for the RS/6000 SP. NIM is a general AIX product, and its introduction to the SP environment was intended to centralize and be coherent with the rest of the RS/6000 products. Most of the NIM commands and options are being hidden by PSSP scripts and commands, so everyone dealing with SP nodes installation and customization must deal with NIM.

In this section we cover some of the problems you can face when dealing with NIM during the installation and customization process.

The Network Install Management (NIM) environment is defined on two characteristic machine roles: *master* and *client*.

NIM Master This role is dedicated to only one machine in the NIM environment. The NIM master is the single point of administration for NIM installations. All NIM-related operations are performed on that machine. Within the RS/6000 SP environment, the machine in the master role is always the CWS.

NIM Client Systems which become installed from the NIM master have the role of NIM client. They remain NIM clients unless this role is explicitly removed. Within the RS/6000 SP environment, nodes were installed as *standalone machines* and do not stay NIM clients.

NIM provides two installation types, depending on whether the NIM master or NIM client is initiating the installation process: pull installation and push installation. Within the RS/6000 SP environment, only pull installation is supported.

The NIM concept is modeled by NIM objects. There are several *classes* of NIM objects on which different sets of operations can be performed. The following operations are defined for all kinds of NIM objects:

- define (creates new NIM objects)
- change (adds new or changes existing attributes of NIM objects)
- remove (removes existing NIM objects)

The different NIM object classes are defined as follows:

Network Objects

The network object describes the network that can be used for the installation process. Several network objects can exist at the same time. Within the RS/6000 SP environment, the network objects name is spnet.

```
# lsnim -l spnet
spnet:
   class      = networks
   type       = ent
   net_addr   = 192.168.0.0
   snm        = 255.255.255.0
   Nstate     = ready for use
   prev_state = information is missing from this object's definition
```

This network object allows the installation of a node by use of the RS/6000 SP internal Ethernet.

Machine Objects

The representation of the systems which become installed are machine objects.

The three types of machine objects are:

- standalone
- diskless
- dataless

NIM client and other information is stored on the master. Within the RS/6000 SP environment, there is a machine object for each node of the RS/6000 SP. They will run as *.standalone* systems after the installation.

```
# lsnim -l sshps01
sphps01:
   class         = machines
   type          = standalone
   Cstate        = ready for a NIM operation
   prev_state    = BOS installation has been enabled
   Mstate        = currently running
   cpuid         = 002028747000
   platform      = rs6k
   if1           = spnet speth01.aixedu 02608CE87C5F ent
   cable_type1   = bnc
   Cstate_result = reset
```

Resource Objects

This class of objects represents resources which are available in the NIM environment. Resource objects represent files and directories which are necessary for some NIM operation like the BOS installation of a NIM client. Several of these resources are allocated to a machine object depending on the software or the used installation process.

There are several types of resource objects.

- lpp_source
- spot
- script
- mksysb

- bosinst_data
- and so on

The complete list of resources is available in the *NIM Installation Guide and Reference*, SC23-2627.

Within the RS/6000 SP environment, the following resource objects are created:

- lppsource (type lpp_source)

  This resource points to the LPPs which are located in the directory /spdata/sys1/install/lppsource in the RS/6000 SP environment.

```
# lsnim -l lppsource
lppsource:
   class       = resources
   type        = lpp_source
   server      = master
   location    = /spdata/sys1/install/lppsource
   alloc_count = 1
   Rstate      = ready for use
   prev_state  = unavailable for use
   simages     = yes
```

- psspspot (type spot)

```
# lsnim -l psspspot
psspspot:
   class        = resources
   type         = spot
   version      = 04
   server       = master
   location     = /usr
   alloc_count  = 1
   Rstate       = ready for use
   prev_state   = ready for use
   release      = 01
   if_supported = rs6k ent
   if_supported = rs6k fddi
   if_supported = rs6k tok
   if_supported = rs6ksmp ent
   if_supported = rs6ksmp tok
   if_supported = rspc ent
   if_supported = rspc tok
```

- noprompt (type bosinst_data)

  The noprompt resource defines some of the installation parameters like the target disks, the installation method (overwrite/migrate) and whether the TCB (Trusted Computing Base) is used.

```
# lsnim -l noprompt
noprompt:
   class       = resources
   type        = bosinst_data
   server      = master
   location    = /spdata/sys1/install/pssp/bosinst_data
   alloc_count = 1
   Rstate      = ready for use
   prev_state  = unavailable for use
```

  Here is an example for the noprompt resource which is defined by the file /spdata/sys1/install/pssp/bosinst_data within the RS/6000 SP environment:

```
# cat /spdata/sys1/install/pssp/bosinst_data
control_flow:
    CONSOLE = /dev/tty0
    INSTALL_METHOD = overwrite
    PROMPT = no
    EXISTING_SYSTEM_OVERWRITE = yes
    INSTALL_X_IF_ADAPTER = no
    RUN_STARTUP = no
    RM_INST_ROOTS = yes
    ERROR_EXIT =
    CUSTOMIZATION_FILE =
    TCB = no
    INSTALL_TYPE = full
    BUNDLES =

target_disk_data:
    LOCATION =
    SIZE_MB =
    HDISKNAME = hdisk0

locale:
    BOSINST_LANG = en_US
    CULTURAL_CONVENTION = en_US
    MESSAGES = en_US
    KEYBOARD = en_US
```

- psspscript (type script)

  The script /spdata/sys1/install/pssp/pssp_script is called at the end of the installation process of a node. It performs, for instance, the following tasks:

  – Configuration of the network adapters except the css0 adapter.

  – Install additional lpps

  – Call the /tftpboot/tuning.cust script

  – Call the /tftpboot/script.cust script

```
# lsnim -l psspscript
psspscript:
   class       = resources
   type        = script
   alloc_count = 1
   server      = master
   location    = /spdata/sys1/install/pssp/pssp_script
   Rstate      = ready for use
   prev_state  = unavailable for use
```

- mksysb_1 (type mksysb)

  This resource defines the location and name of the image to be installed on the nodes. There is one mksysb resource for each installation image on the CWS.

```
# lsnim -l mksysb_1
mksysb_1:
   class       = resources
   type        = mksysb
   alloc_count = 1
   server      = master
   location    = /spdata/sys1/install/images/bos.obj.ssp.41
   Rstate      = ready for use
   prev_state  = unavailable for use
   version     = 4
   release     = 1
```

Before you install, customize, maintain, or diagnose a node, it is necessary to allocate the appropriated NIM resource to the machine object. This resource allocation is done by the *setup_server* script by calling several NIM commands. Following is an example of a machine object, called sp21n01, that has been enabled to be installed. The corresponding allocations have been highlighted to reflect the additional attributes of the NIM machine object.

```
# lsnim -l sp21n01
sp21n01:
   class        = machines
   type         = standalone
   Cstate       = BOS installation has been enabled
   prev_state   = ready for a NIM operation
   Mstate       = currently running
   cpuid        = 002028747000
   platform     = rs6k
   if1          = spnet speth01.aixedu 02608CE87C5F ent
   cable_type1  = bnc
   spot         = psspspot
   control      = master
   lpp_source   = lppsource
   bosinst_data = noprompt
   script       = psspscript
   mksysb       = mksysb_1
   boot         = boot
   nim_script   = nim_script
```

2.9 NIM Commands

In some cases, the configuration of the NIM master does not work as smoothly as displayed so far. In these cases, you have to perform some diagnostic tasks. NIM offers you a set of commands you can run to do this. A list of some of them follows:

lsnim This command displays information about the NIM environment, including predefined information, the attributes required for a NIM Operation, all customized NIM Objects, information on specific NIM objects and information about resources available to specific NIM machines.

nim The `nim` command performs operations on NIM objects. The type of operation performed is dependent on the type of object on which this operation is performed. This means that the type of operation is different, for instance, on machine and resource objects.

nimclient The `nimclient` command is used by workstations that are NIM clients to pull NIM resources. This command can enable or disable the NIM master server's ability to initiate workstation installation and customization. In the RS/6000 SP environment, this command is not used. All NIM configurations are performed directly on the CWS.

nimconfig The `nimconfig` command initializes the NIM master package. This includes, for instance, the creation and start of the nimesis process (SRC-controlled) which is used for communication purposes between NIM Master and NIM Clients. In the RS/6000 SP environment, this command is called by the setup_server script.

niminit This command configures the NIM client package. Call this command before you use the command nimclient. If the workstation has been installed from a NIM master, it is already a NIM client and this command is not used. There is no need to use this command in the RS/6000 SP environment.

Let's have a closer look at the lsnim command and the nim command. These two commands are basically used for NIM administration purposes once the NIM master package has been configured.

```
# lsnim
master              machines        master
boot                resources       boot
nim_script          resources       nim_script
spnet               networks        ent
lppsource           resources       lpp_source
psspspot            resources       spot
noprompt            resources       bosinst_data
prompt              resources       bosinst_data
psspscript          resources       script
mksysb_1            resources       mksysb
sphps01             machines        standalone
sphps03             machines        standalone
sphps05             machines        standalone
sphps06             machines        standalone
```

Without any parameters, the command lsnim displays all NIM objects which have been created on this NIM master.

In the RS/6000 SP environment, we see the previously mentioned resources.

The -l option displays the set of attributes, which are associated with a specific NIM object.

```
# lsnim -l sphps01
sphps01:
   class         = machines
   type          = standalone
   Cstate        = ready for a NIM operation
   platform      = rs6k
   if1           = spnet speth01.aixedu 02608CE87C5F ent
   cable_type1   = bnc
   prev_state    = BOS installation has been enabled
   cpuid         = 002028747000
   Mstate        = currently running
   Cstate_result = reset
```

The meanings of the different attributes follow:

class This attribute represents the object class. In addition to machines, there are the classes resource and network.

type The type attribute defines the machine type. In this case the workstation is a standalone system. Other possible states are diskless and dataless. The RS/6000 SP environment supports only the machine type standalone.

Cstate This attribute defines the control state of the machine. The possible states are:

- Ready for NIM operation.

  (Required state to perform an operation on an object.)
- Diskless or dataless boot is enabled.

  (This is not relevant in the RS/6000 SP environment.)
- Diskless resources are being initialized.

  (This is not relevant in the RS/6000 SP environment.)
- Dataless install has been enabled.

  (This is not relevant in the RS/6000 SP environment.)
- Diagnostic boot has been enabled.
- BOS installation has been enabled.
- Base operating system installation is being performed.
- Customization is being performed.
- Post-install process is being performed.
- Software maintenance is being performed.

platform The hardware platform of the system:

- *rs6k* Micro Channel-based uniprocessor systems
- *rs6ksmp* Micro Channel-based symmetric multiprocessor systems
- *rspc* ISA-bus systems

In the RS/6000 SP environment, only *rs6k* is relevant.

if1 This attribute specifies the interface to be used for the installation. It holds the name of the used network resource *spnet*, the later host name of the system, the MAC address of the network adapter that is used for network booting, and the type of network adapter.

cable_type1 This is bnc or dix in the RS/6000 SP environment.

prev_state This attribute displays the previous state of the NIM object.

cpuid Stores the ID of the CPU.

Mstate This attribute defines the *machine state*. Possible values are:

- Currently running
- Not running
- In the process of booting

It is not guaranteed that this state always displays the real state of the system.

Cstate_result This attribute specifies the result of the last control applied to this machine object.

The nim command

The `nim` command is used to perform operations on NIM machine objects.

```
# nim -o <operation> -a <attribute>=<value> ... <object>

# lsnim -O sphps01
sphps01:
   define     = define an object
   change     = change an object's attributes
   remove     = remove an object
   allocate   = allocate a resource for use
   deallocate = deallocate a resource
   diag       = enable a machine to boot a diagnostic image
   cust       = perform software customization
   bos_inst   = perform a BOS installation
   maint      = perform software maintenance
   reset      = reset an object's NIM state
   lslpp      = list LPP information about an object
   fix_query  = perform queries on installed fixes
   check      = check the status of a NIM object
   reboot     = reboot specified machines
```

This -O option of the lsnim command is very useful in determining the set of operations that can be performed on a specific object (the object *sphps01* of type *machine*, in this case).

define This operation creates a new NIM machine object. In the RS/6000 SP environment, it is called by setup_server for all nodes which are known to the SDR.

```
# nim -o define -t standalone -a if1='spnet speth02 10005aa88123' \
    -a cable_type1=dix newhost
```

This operation creates the standalone NIM machine object newhost. The NIM Network object *spnet* must exist at this time. Several checks are made; for instance, whether the IP address for hostname *speth02* is reachable from the net spnet. Furthermore, the attribute cable_type1 is a required for Ethernet network objects. There are other optional attributes for machine objects: iplrom_emu, cpuid, comments, ring_speed, cable_type, and platform. The script setup_server uses the attributes platform and cpuid in addition to the given example.

change This operation changes the values of attributes.

```
# nim -o change -a cable_type1=bnc newhost
```

or

```
# nim -o change -a comments="This is a comment..." newhost
```

To remove an optional attribute, just let the value for the attribute void while changing the object.

```
# nim -o change -a comments= newhost
```

remove This operation removes a NIM machine object. Removing machine objects is successful in case no resources are allocated to that machine object and its Cstate is Ready for a NIM operation. This state can be enforced by resetting the object:

```
# nim -o remove newhost
```

The script setup_server removes and redefines machine objects, for instance, in case some of the node-specific information like the hardware address of the SP Ethernet adapter changes. This

change is done when the next boot responds of the involved node requires a network boot.

allocate This operation allocates some resources to a machine object. Sometimes more then one resource must be allocated before an operation like bos_inst or diag can be performed. The operation diag, for instance, requires a spot resource to be allocated first.

```
# nim -o allocate -a spot=psspspot newhost
```

deallocate This operation deallocates resources from a machine object. Each resource owns an allocation counter specifying how often this resource has been allocated by machine objects. These resources can be removed only when their allocation counter has the value 0. This means the resource is not allocated by any machine object.

```
# nim -o deallocate -a spot=psspspot newhost
```

diag This operation performs all configurations necessary to enable a diagnostic image to be booted at the next netboot time. In the RS/6000 SP environment this is done by:

```
# spbootins -r diag -l 3
```

which changes the boot responds value for node number 3 and calls setup_server, which allocates the necessary resources (psspspot) and calls:

```
# nim -o diag sphps03
```

cust This operation is used to install and update software or to execute scripts on standalone clients. In the RS/6000 SP environment this operation is not used. It requires the nodes to be configured as NIM Clients (access to root user granted by entry in .rhosts file on the node). The following example shows how to install the Performance Agent software on a workstation. First an lpp_source must be allocated to the machine object on which the software is to be installed. The lpp_source must contain the fileset being installed. Then the installation is performed in the second step:

```
# nim -o allocate -a lpp_source=lppsource newhost
# nim -o cust \
      -a fileset="perfagent"
      -a installp_flags="-agX" newhost
```

The *lpp_source* resource is deallocated automatically thereafter.

bos_inst The operation *bos_inst* enables a workstation being installed from a NIM master. Before this operation can be performed successfully, some resources have to be allocated to the machine object of that workstation. The resources spot and lpp_source are required. The resources bosinst_data, image_data, installp_bundle, mksysb, and script are optional.

```
# nim -o bos_inst newhost
```

The script setup_server uses this operation to setup the installation for RS/6000 SP nodes. Here the additional attributes no_client_boot and source are used.

```
# nim -o bos_inst -a source=mksysb -a no_client_boot=yes sphps03
```

The following list of resources has been allocated prior to the bos_inst operation in the RS/6000 SP environment: psspspot, lppsource, noprompt, psspscript and mksysb_1.

maint This operation is used to perform operations on installed software, such as removing or cleaning up after an interrupted installation. *It is not used in the RS/6000 SP environment because nodes are not configured as NIM Clients.* The following example shows how to uninstall the Performance Agent software from a workstation.

```
# nim -o maint -a installp_flags="-u" -a filesets=perfagent newhost
```

reset This operation is used to reset the Cstate of a machine object back to *Ready for a NIM operation*. Resources are not automatically deallocated. The following example resets the machine object newhost.

```
# nim -o reset newhost
```

lslpp This operation is used to display all LPPs installed on a workstation. It is not used in the RS/6000 SP environment because it would require the nodes to be configured as NIM Clients.

```
# nim -o lslpp newhost
```

fix_query NIM provides the fix_query operation as an interface to the instfix command to provide the ability to list information about installed fixes. The *fix_query* operation can be applied to SPOT resources or NIM clients. When invoked without any optional attributes, information about all installed fixes are displayed on the target. The fixes attribute and fix_bundle resource are available to the *fix_query* operation to install particular fixes. This operation is not used in the RS/6000 SP environment because it would require the nodes to be configured as NIM Clients.

```
# nim -o fix_query newhost
```

check The operation *check* performs some basic consistency checks on the given machine object.

```
# nim -o check newhost
```

reboot This operation is used to reboot workstations. It is not used in the RS/6000 SP environment because it would require the nodes to be configured as NIM Clients.

```
# nim -o reboot newhost
```

The following list of operations is used by the setup_server script: *define*, *change*, *remove*, *allocate*, *bos_inst*, and *diag*.

To Start All over Again

If you need to start NIM completely over, you can use this set of commands:

```
# nim -o unconfigure master
# installp -u bos.sysmgt.nim.master
# setup_server
```

To Reset the Machine State

Sometimes you need to reset the state of a machine object after a failed installation or customization. You can do it in this way:

```
# nim -Fo reset <machine object>
# nim -o deallocate -a subclass=all <machine object>
```

Or you can do it by using PSSP commands:

```
# spbootins -r disk -l <node number> ...
```

2.9.1 NIM Diagnostics

Because NIM maintains its own information about the state of the nodes, separated from the SDR information, you may get inconsistences between these two SP components, especially when you are trying to install or customize SP nodes, and NIM does not allocate resources to it. Here is a list of steps you can follow to check that all the NIM information is consistent with the SDR information:

- Reviewing NIM Client Definition:

  1. Determine the client's NIM master by using:

     `splstdata -b -l <client_node_number>`

     Look at the srvr field, which lists the NIM master's node_number.

```
# splstdata -b -l 3
                List Node Boot/Install Information

Node#        hostname  hdw_enet_addr srvr     response            install
        last_install_image    last_install_time   next_install_image
------------------------------------------------------------------------
   3 sp2n03           10005AFA082C    1         disk              hdisk0
          bos.obj.ssp.41  Sat_May__4_12:57:38       bos.obj.ssp.41
```

  2. Determine the NIM master hostname by issuing `splstdata -b -l <server_node_number>` and looking at the reliable host name.

```
# splstdata -b -l 1
                 List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response            instal
         last_install_image    last_install_time   next_install_image
------------------------------------------------------------------------
    1 sp2n01             10005AFA18CF    0         disk
            bos.obj.ssp.41  Sat_May__4_10:42:40       bos.obj.ssp.41
```

  3. Log in to the NIM master by using the `telnet` or `rsh` command:

     `# telnet sp2n01`

  4. Use `lsnim` to list the objects in the NIM master's database:

```
# lsnim
master         machines        master
boot           resources       boot
nim_script     resources       nim_script
spnet          networks        ent
sp2cw0         machines        standalone
lppsource      resources       lpp_source
psspspot       resources       spot
noprompt       resources       bosinst_data
prompt         resources       bosinst_data
```

```
psspscript     resources       script
mksysb_1       resources       mksysb
cw_tr0         networks        tok
sp2n02         machines        standalone
sp2n03         machines        standalone
sp2n04         machines        standalone
sp2n05         machines        standalone
sp2n06         machines        standalone
sp2n07         machines        standalone
sp2n08         machines        standalone
```

5. While still on NIM master, issue `lsnim -l <client_name>` to list the NIM client definition for the node that is having the problem:

```
# lsnim -l sp2n03
sp2n03:
   class         = machines
   type          = standalone
   Cstate        = ready for a NIM operation
   Mstate        = currently running
   Cstate_result = success
   prev_state    = not running
   platform      = rs6k
   if1           = spnet sp2n03 10005AFA082C ent
   cable_type1   = bnc
   cpuid         = 000116875700
```

6. From the first step (output of `splstdata -b -l 3`), node 3 (sp2n03) had a response (bootp_response) of *disk*. The `Cstate` attribute for the client *Ready for nim operation* from the output of `lsnim -l sp2n03` is correct.

Based on the bootp_response for the node, the Cstate attribute is fixed and will have fixed NIM object allocations for the client. The following table could be used as a ready reckoner for NIM client definition information.

bootp_response	**Cstate**	**Allocation**	
install	BOS installation has been enabled	**spot**	psspspot
		lpp_source	lppsource
		bosinst_data	noprompt
		script	psspscript
		mksysb	mksysb_1
diag	Diagnostic boot has been enabled	**spot**	psspspot
		bosinst_data	prompt
maintenance	BOS installation has been enabled	**spot**	psspspot
		bosinst_data	prompt
disk or customize	Ready for a NIM operation		

Table 5. NIM Client Definition Information

2.9.1.1 Export Problems

Following is an example of a successful export by the exportfs command from NIM master node sp2n01 of a NIM client sp2n08, which is ready to be installed.

```
# exportfs sp2n03
/var/adm/acct                                -root=SP2CW0:        \
  sp2tr0:sp2cw0,access=SP2CW0:sp2tr0:sp2cw0=machines
/spdata/sys1/install/pssplpp                 -ro
/export/nim/scripts/sp2n08.script            -ro,root=sp2n08,access=sp2n08
/usr                                         -ro,root=sp2n08,access=sp2n08
/spdata/sys1/install/pssp/bosinst_data       -ro,root=sp2n08,access=sp2n08
/spdata/sys1/install/pssp/pssp_script        -ro,root=sp2n08,access=sp2n08
/spdata/sys1/install/images/bos.obj.ssp.41 -ro,root=sp2n08,access=sp2n08
```

However, sometimes a node will appear in *exportfs* output from the NIM master even if you never defined that particular node as a NIM client. For example, this could happen if the NIM database is out of sync in a situation where NIM has not successfully removed the NIM client with a previous NIM command.

To diagnose the problem with the definition of your nodes as NIM clients, use the exportfs command to list the exported directories on the NIM master.

1. List the exports:

   ```
   # exportfs
   /spdata/sys1/install/pssplpp                -ro
   /spdata/sys1/install/lppsource              -ro
   /spdata/sys1/install/images/bos.obj.ssp.41  -ro,root=sp2n08
   ```

   In this case, assume *sp2n08* is the client that you are trying to define, but the exportfs command does not list resources that should be allocated to the NIM client. At the same time *sp2n08* is listed in the /spdata/sys1/install/images/bos.obj.ssp.41 entry.

2. To correct this situation, you must issue the rmnfsexp command to remove the client from the exported list.

   ```
   # rmnfsexp -d /spdata/sys1/install/images/bos.obj.ssp.41
   unexported /spdata/sys1/install/images/bos.obj.ssp.41
   ```

3. List the exportfs again to verify that the exported directory has been removed from the export list:

   ```
   # exportfs
   /spdata/sys1/install/pssplpp                -ro
   /spdata/sys1/install/lppsource              -ro
   ```

   Now that the NFS export has been corrected, you can issue setup_server on the NIM master to redefine the NIM client.

2.9.2 Setup_server Fails

There are several reasons why setup_server can fail while creating and allocating NIM resources. The most likely case is that the lppsource, psspspot, or mksysb_1 resource cannot be created. It is quite unlikely that one of the resources noprompt, prompt, or psspscript fails to create.

The noprompt resource needs access to the file /spdata/sys1/install/pssp/bosinst_data. This file is copied by setup_server from the file /usr/lpp/ssp/install/config/bosinst_data.template. Make sure this file exists before setup_server is called. This file is part of the ssp.basic fileset.

The same applies to the resources prompt and psspscript. Make sure the files /usr/lpp/ssp/install/config/bosinst_data_prompt.template and /usr/lpp/ssp/install/bin/pssp_script exist.

2.9.2.1 lpp_source Creation Fails

```
# setup_server

setup_server: Creating the lppsource resource
warning: 0042-176 c_mk_lpp_source: The resource located in
        /spdata/sys1/install/lppsource
        cannot serve as a "support images" (or "simages") lpp_source
        because one or more of the following filesets are missing:
                bos.net


setup_server: 0016-072 Error detected processing the following nim command:
/usr/sbin/nim -o define -t lpp_source -a server=master \
  -a location=/spdata/sys1/ install/lppsource lppsource
Return code = 8. Check any previous error messages for possible problems.
```

There are several reasons why the SP lpp_source *lppsource* can fail to be created. The most likely reason for this is that some filesets are missing in the /spdata/sys1/install/lppsource directory. Compare all filesets in this directory with the list given in *Step 8* of the installation process in the *RS/6000 SP Installation Guide*, GC23-3898.

This list is defined in the file /usr/lpp/bos.sysmgt/nim/methods/c_sh_lib by the variable SIMAGES_OPTIONS

```
SIMAGES_OPTIONS="\
        bos \
        bos.info.any \
        bos.net \
        bos.rte.up \
        bos.rte.mp \
        bos.diag \
        bos.powermgt \
        bos.sysmgt \
        bos.terminfo.all \
        devices.all \
        X11.apps \
        X11.base \
        X11.compat \
        X11.Dt \
        X11.fnt \
        X11.loc \
        X11.motif \
        X11.msg.all \
        X11.vsm"
```

In this case, the fileset bos.net is missing. To recover from the problem, copy the missing fileset into /spdata/sys1/install/lppsource and run setup_server again.

2.9.2.2 spot Creation Fails

If setup_server fails to create the psspspot resource, verify that the following resources are available:

1. First verify that none of the filesystems are full.
2. Verify that the root filesystem (/) has enough space. The NIM network install images will reside in /tftpboot. NIM recommends that you have enough

room for eight 2.5 MB images or a total of 20 MB. However, once the images have been created, you only need psspspot.rs6k.ent. The other images could be deleted. Accordingly, one may say that you only need one file of 2.5 MB.

3. There should be at least 1 MB of free space in /tmp.
4. List the lppsource resource to see if it is available.

```
# lsnim -l lppsource
lppsource:
   class       = resources
   type        = lpp_source
   server      = master
   location    = /spdata/sys1/install/lppsource
   alloc_count = 0
   Rstate      = ready for use
   prev_state  = unavailable for use
   simages     = yes
```

The *Rstate* is *ready for use* and the simages is yes.

If the simages attribute is no, then the required images for the support images needed to create the SPOT were not available in the *lppsource* resource. The required images needed for the SPOT creation are listed in the REQUIRED_SIMAGES variable in the file /usr/lpp/bos.sysmgt/nim/methods/c_sh_lib.

```
REQUIRED_SIMAGES="\
        bos \
        bos.net \
        bos.rte.up \
        bos.rte.mp \
        bos.diag \
        bos.sysmgt \
        bos.terminfo \
        bos.terminfo.all.data \
        devices.base.all \
        devices.buc.all \
        devices.graphics.all \
        devices.mca.all \
        devices.scsi.all \
        devices.sio.all \
        devices.sys.all \
        devices.tty.all"
```

Before you run setup_server, again it may be necessary to remove the incomplete created psspspot.

```
# nim -o remove psspspot
```

2.9.2.3 mksysb Creation Fails

```
# setup_server

0042-001 nim: processing error encountered on "master":
   0042-001 m_mkbosi: processing error encountered on "master":
   0042-154 c_stat: the file "/spdata/sys1/install/images/bos.obj.ssp.41"
does not exist

setup_server: 0016-072 Error detected processing the following nim command:
/usr/sbin/nim -o define -t mksysb -a server=master -a location=/spdata/sys1/inst
all/images/bos.obj.ssp.41 mksysb_1
Return code = 1. Check any previous error messages for possible problems.
```

If setup_server cannot create the mksysb_1 resource, verify that:

1. /spdata/sys1/install/images exists with permissions rwxr-sr-x.
2. The mksysb image exists with the correct name, either the default mksysb name or the one you have specified before running setup_server. Make sure the permissions are at least rw-r--r--.

If you have more than one mksysb image, the names of the mksysb resources are mksysb_2, mksysb_3, and so on.

2.9.2.4 psspspot Allocation Fails

```
# setup_server

setup_server: Checking NIM client - sphps01
setup_server: Checking NIM client allocations - sphps01.aixedu
setup_server: Allocating resources for client sphps01
0042-001 nim: processing error encountered on "master":
   0042-001 m_allocate: processing error encountered on "master":
   0042-058 m_alloc_spot: unable to allocate "psspspot" to "sphps01"
        because it does not support the network interface type
        of that client

setup_server: 0016-072 Error detected processing the following nim command:
/usr/sbin/nim -o allocate -a spot=psspspot -a lpp_source=lppsource -a bosinst_da
ta=noprompt -a script=psspscript -a mksysb=mksysb_2 sphps01
Return code = 1. Check any previous error messages for possible problems.
```

If allocating the psspspot resource fails, follow these steps to determine and correct the problem:

1. Perform a check on the SPOT

   ```
   # nim -o check psspspot
   ```

   This check should inform you if there is a problem. In our environment the command produced the following message:

   ```
   0042-001 nim: processing error encountered on "master":
      0042-062 m_ckspot: "psspspot" is missing something which is
           required
   ```

2. If you are unable to determine the problem with the SPOT, you can update the SPOT by issuing the following command:

   ```
   # nim -o cust -a lpp_source=lppsource psspspot
   ```

   In our environment this did not work either because of the severity of the problem with the SPOT. Instead we got the following message:

```
0042-001 nim: processing error encountered on "master":
   0042-001 m_instspot: processing error encountered on "master":
   0042-062 m_ckspot: "psspspot" is missing something which is
        required
```

3. If the problem persists, you can remove the SPOT.

   ```
   # nim -Fo remove psspspot
   ```

   Then run setup_server to recreate the SPOT. Alternatively, you can run the following nim command:

   ```
   # nim -o define -t spot -a server=master -a source=lppsource \
     -a location=/usr psspspot
   ```

2.9.3 Debugging NIM Installations

Sometimes, the only way to discover a problem is by debugging NIM. This procedure will help to precisely identify which step in the installation process is failing. However, the actions needed to correct the problem will vary depending on the failing step. The debug option is a low lever tool provided by creating a spot in debug mode, which will let you run the installation process in a step-by-step basis. Perform the following steps to debug the installation of your nodes.

1. `spbootins -r disk <Frame#> <Node#> <NumberOfNodes>`

   This command resets the specified nodes to *disk* and calls all necessary NIM commands to reset the install configuration. Always use this command to start your new configuration from a well-defined point.

2. `nim -Fo check -a debug=yes psspspot`

   This command creates the SPOT in debug mode.

3. `lsnim -l psspspot`

   Display the attributes of the SPOT resource and look for a line containing the string enter_debug = rs6k. There will be an address imbedded in the text like 0x0013d410. Remove the starting 0x and write the remaining number down. It will be used in a later step.

```
# lsnim -l psspspot

psspspot:
   class        = resources
   type         = spot
   version      = 04
   server       = master
   location     = /usr
   alloc_count  = 0
   Rstate       = ready for use
   prev_state   = ready for use
   release      = 01
   if_supported = rs6k ent
   if_supported = rs6k fddi
   if_supported = rs6k tok
   if_supported = rs6ksmp ent
   if_supported = rs6ksmp tok
   if_supported = rspc ent
   if_supported = rspc tok
   enter_dbg    = "rs6k 0x0013d410"
   enter_dbg    = "rs6ksmp 0x0015d2b0"
   enter_dbg    = "rspc 0x0013d410"
```

4. `spbootins -r install <Frame#> <Node#> <NumberOfNodes>`

All the specified nodes are set to *install* and the NIM commands for resource allocation are called.

5. `nim -Fo check psspspot`

   This last command resets the SPOT back to a no debug version. Do not forget to do this step unless you have also installation problems on other nodes.

2.10 Network Booting

There are three different booting modes supported by PSSP V2.1 which require a network boot to be performed.

1. Installing nodes

   Typically no external devices like tape drives or CDROM drives are connected to the nodes of a RS/6000 SP. Therefore, it is necessary to install the nodes over the network, including the boot phase. This also applies to the two following tasks.

2. Maintaining nodes

   Normally, the *maintenance* mode is used when there is a problem booting the node, and it is necessary to get access to the *rootvg* volume group to figure out and fix the problem. More details about this mode will be given later.

3. Diagnosing nodes

   Using this mode allows you to run some diagnostics on any device, including disks and SCSI adapters, since the node is booted from the network. This mode will be covered later in this chapter in more detail.

The command `spbootins` prepares the installation server to be able to serve NIM installations, including serving as network boot server. A variety of other task are also performed by `spbootins`. For instance, the bootp_response field for the node in the SDR is changed, and the script setup_server is called. The configuration of the installation server itself is done by setup_server.

The following example prepares the boot/install server to install the node with node numbers 1 and 3.

```
# spbootins -r install -l 1,3
```

The spbootins scripts supports the following boot response modes:

1. install

   With this mode, `setup_server` will allocate the following NIM resources: *psspspot*, *lppsource*, *noprompt*, *psspscript* and *mksysb_1*. Furthermore the files /tftpboot/<hostname>.info, /tftpboot/<hostname>-new-srvtab, /tftpboot/<hostname>.install_info and /tftpboot/<hostname>.config_info are created and /etc/bootptab is updated to enable this machine to act as boot server. The bootp_response *install* requires a network boot.

2. maintenance

   In case you select the maintenance bootp_response, other resources are allocated to the NIM client representing the node which will be run in maintenance mode: psspspot, lppsource, prompt, boot, and nim_script.

Similar to the install mode the files /tftpboot/<hostname>.info, /tftpboot/<hostname>.install_info, and /tftpboot/<hostname>.config_info are created and /etc/bootptab is updated to enable this machine to act as boot server. The bootp_response maintenance requires a network boot.

3. diag

   The selection of the *diag* bootp_response allocates the following resources to the NIM machine object: psspspot, prompt, and boot. The same file as in the two other boot response modes are being created. The bootp_response *diag* requires a network boot.

4. customize

   The bootp_response *customize* deallocates all resources of a given NIM client, removes the NIM client, and recreates it. No resources are allocated. The files /tftpboot/<hostname>.install_info and /tftpboot/<hostname>.config_info are created beside /tftpboot/<hostname>-new-srvtab. No network boot is required.

5. disk

   This bootp_response mode finally changes the bootp_response field in the SDR to *disk* and removes the resource allocations of a NIM client. The file /tftpboot/<hostname>.info is removed, as well as the entry in the /etc/bootptab file for that client. No network boot is required.

2.10.1 Scripts Involved in Network Boot

The main script involved in the network boot process is /sbin/rc.boot. This script is called first in boot phase one (/sbin/rc.boot 1). Basically the following steps are performed at that time:

1. Read the IPL control block (`bootinfo -b`) to get information such as client address, the boot server address, gateway information, the network adapter to use, the bootfile name, and others.

2. Configure the given network device by use of the received IP parameters. If this fails, you see LED code **607**, a failure that is quite unlikely.

3. Now the /tfptboot/<hostname>.info file is tftp'd from the installation server to the local file /SPOT/niminfo. When this tftp transfer fails, the client node stops with a LED code of **608**. This file contains a set of environment variables which define the current installation environment. This file is sourced (/SPOT/niminfo) and the environment variables are set in the current shell. If this fails, you see the LED error code **609**.

4. A more likely step to fail is the mount which is performed now. The SPOT is mounted in the RS/6000 SP environment by the command `mount bootserver:/usr /SPOT/usr`. This step sometimes fails due to incorrect NFS configuration on the installation server. The typical LED error code is **611**.

   Sometimes, there is a routing problem between the client node and the Control Workstation. Even when the client node is installing from a boot/install server other than the Control Workstation, a route is required from the Control Workstation to the client node. If this is the case, **610** is the LED code where the node will stop. If you see the **612**, the mount was successful. We discuss this situation later in this chapter.

5. Depending on the reason for the network boot, the next step may vary. If you perform an installation, the script /usr/lib/boot/network/rc.bos_inst is copied from the SPOT into the local /etc: directory and also sourced. If you

perform a network boot for diagnostic reasons, the file /usr/lib/boot/network/rc.diag is copied and sourced. The last possible script is /usr/lib/boot/network/rc.dd_boot, but it is not used in the RS/6000 SP environment. It would perform a normal network boot (for instance, in a diskless/dataless environment).

The /etc/rc.bos_inst script also distinguishes boot phases.

In the first phase, some filesystems and files are mounted from the installation server. This list of directories is defined in the NIM_MOUNTS environment variable, set by the script /SPOT/niminfo.

This list includes the directory /spdata/sys1/install/lppsource, which is mounted over /SPOT/usr/sys/inst.images; the file /spdata/sys1/install/pssp/bosinst_data, which is mounted over /NIM_BOSINST_DATA, and the mksysb file /spdata/sys1/install/images/bos.obj.ssp.41, which is mounted on /NIM_BOS_IMAGE.

In case one of these mounts cannot be performed, you will see the LED error code **611**.

The /usr/lpp/bosinst/bi_main script is called from /etc/rc.bos_inst during the second phase of /sbin/rc.boot. This script triggers the installation process.

If you have problems within the beginning of the installation, look at some of these scripts to figure out which condition may cause the installation to fail.

Let's have a closer look at the /tftpboot/<hostname>.info file to gain a better understanding of some parameters of the installation process. This will help us to find and fix network boot and installation-related problems.

2.10.1.1 The /tftpboot/<hostname>.info File

```
export NIM_NAME=sphps03

export NIM_MASTER_HOSTNAME=spcntl.aixedu

export RC_CONFIG=rc.bos_inst

export SPOT=spcntl.aixedu:/usr

export NIM_BOSINST_DATA=/NIM_BOSINST_DATA

export NIM_BOS_IMAGE=/NIM_BOS_IMAGE

export NIM_BOS_FORMAT=mksysb

export NIM_MOUNTS=" spcntl.aixedu:/spdata/sys1/install/lppsource: \
  /SPOT/usr/sys/inst.images:dir \
  spcntl.aixedu:/spdata/sys1/install/pssp/bosinst_data:/NIM_BOSINST_DATA:file \
  spcntl.aixedu:/spdata/sys1/install/images/bos.obj.ssp.41:/NIM_BOS_IMAGE:file "
```

This is only an excerpt of some important variables of the /tftpboot/<hostname>.info file created for a installation of a mksysb image.

NIM_NAME

This variable defines the name of the NIM client object owning this file.

NIM_MASTER_HOSTNAME

This variable defines the name of the NIM master from which this NIM client is installed.

RC_CONFIG

This variable contains the name of the script used to continue the installation. This script will be called from the /etc/rc.boot script. Possible names other than rc.bos_inst are rc.diag and rc.dd_boot.

SPOT

This variable defines the location of the SPOT on the boot/installation server. In the PSSP 2.1 environment, this is always /usr.

NIM_BOSINST_DATA

The name of the bosinst_data file for this installation is stored in this environment variable.

NIM_BOS_IMAGE

Here we find the name of the image to be installed on the node.

NIM_BOS_FORMAT

One format of an installation image is `mksysb`. Another is, for instance, *rte*, which leads to a basic BOS installation. In the RS/6000 SP environment, this variable is always set to `mksysb` for the installation process. In case of a diagnostic network boot, it contains the value `rte`. Compare this with the three examples given following this list.

NIM_MOUNTS

This variable defines the list of directories and files being mounted from the installation server while performing the requested network boot process.

Following is an example of a host.info file created for a NIM installation:

```
#------------------ Network Install Manager ---------------
# warning - this file contains NIM configuration information
# and should only be updated by NIM
export NIM_NAME=sphps03
export NIM_HOSTNAME=speth03.aixedu
export NIM_CONFIGURATION=standalone
export NIM_MASTER_HOSTNAME=spcntl.aixedu
export NIM_MASTER_PORT=1058
export RC_CONFIG=rc.bos_inst
export NIM_BOSINST_ENV="/../SPOT/usr/lpp/bos.sysmgt/nim/methods/c_bosinst_env"
export NIM_BOSINST_RECOVER="/../SPOT/usr/lpp/bos.sysmgt/nim/methods/ \
  c_bosinst_env -a hostname=speth03.aixedu"
export SPOT=spcntl.aixedu:/usr
export NIM_BOSINST_DATA=/NIM_BOSINST_DATA
export NIM_CUSTOM="/../SPOT/usr/lpp/bos.sysmgt/nim/methods/ \
c_script -a location=spcntl.aixedu:/export/nim/scripts/sphps03.script"
export NIM_BOS_IMAGE=/NIM_BOS_IMAGE
export NIM_BOS_FORMAT=mksysb
export NIM_HOSTS=" 192.168.0.3:speth03.aixedu  192.168.0.200:spcntl.aixedu "
export NIM_MOUNTS=" spcntl.aixedu:/spdata/sys1/install/lppsource: \
 /SPOT/usr/sys/inst.images:dir \
 spcntl.aixedu:/spdata/sys1/install/pssp/bosinst_data:/NIM_B OSINST_DATA:file \
 spcntl.aixedu:/spdata/sys1/install/images/bos.obj.ssp.41:/N IM_BOS_IMAGE:file "
```

Following is an example of a host.info file created for a maintenance network boot:

```
#------------------ Network Install Manager ---------------
# warning - this file contains NIM configuration information
# and should only be updated by NIM
export NIM_NAME=sphps03
export NIM_HOSTNAME=speth03.aixedu
export NIM_CONFIGURATION=standalone
export NIM_MASTER_HOSTNAME=spcntl.aixedu
export NIM_MASTER_PORT=1058
export RC_CONFIG=rc.bos_inst
export NIM_BOSINST_ENV="/../SPOT/usr/lpp/bos.sysmgt/nim/methods/c_bosinst_env"
export NIM_BOSINST_RECOVER="/../SPOT/usr/lpp/bos.sysmgt/nim/methods/ \
  c_bosinst_env -a hostname=speth03.aixedu"
export SPOT=spcntl.aixedu:/usr
export NIM_BOSINST_DATA=/NIM_BOSINST_DATA
export NIM_CUSTOM="/../SPOT/usr/lpp/bos.sysmgt/nim/methods/ \
c_script -a location=spcntl.aixedu:/export/nim/scripts/sphps03.script"
export NIM_BOS_IMAGE=/SPOT/usr/sys/inst.images/bos
export NIM_BOS_FORMAT=rte
export NIM_HOSTS=" 192.168.0.3:speth03.aixedu  192.168.0.200:spcntl.aixedu"
export NIM_MOUNTS=" spcntl.aixedu:/spdata/sys1/install/lppsource: \
/SPOT/usr/sys/inst.images:dir \
spcntl.aixedu:/spdata/sys1/install/pssp/bosinst_data_prompt: \
/NIM_BOSINST_DATA:file"
```

Following is an example of a host.info file created for a diagnosis network boot:

```
#------------------ Network Install Manager ---------------
# warning - this file contains NIM configuration information
# and should only be updated by NIM
export NIM_NAME=sphps03
export NIM_HOSTNAME=speth03.aixedu
export NIM_CONFIGURATION=standalone
export NIM_MASTER_HOSTNAME=spcntl.aixedu
export NIM_MASTER_PORT=1058
export RC_CONFIG=rc.diag
export SPOT=spcntl.aixedu:/usr
export NIM_BOSINST_DATA=/NIM_BOSINST_DATA
export NIM_BOS_IMAGE=/SPOT/usr/sys/inst.images/bos
export NIM_BOS_FORMAT=rte
export NIM_HOSTS=" 192.168.0.3:speth03.aixedu  192.168.0.200:spcntl.aixedu "
export NIM_MOUNTS=" spcntl.aixedu:/spdata/sys1/install/pssp/ \
  bosinst_data_prompt:/NIM_BOSINST_DATA:file "
```

2.10.2 Node Installation Diagnostics

If a node fails to install a mksysb image from its boot/install server in network boot mode, then try the node conditioning method. Node conditioning may provide you with information as to where the node install is failing.

The following steps can be used to determine the cause of the node install failure and how to resolve them.

Step 1: Verify Boot/Install Server is Available

Do the following steps to verify that the boot/install server is available:

1. Determine the client's boot/install server by issuing: `splstdata -b -l <client_node_number>` and looking at the srvr field, which lists the boot/install server's node_number.

```
# splstdata -b -l 3
                List Node Boot/Install Information

Node#        hostname  hdw_enet_addr srvr     response            install
        last_install_image    last_install_time   next_install_image
------------------------------------------------------------------------
   3 sp2n03           10005AFA082C    1         disk              hdisk0
         bos.obj.ssp.41  Sat_May__4_12:57:38       bos.obj.ssp.41
```

2. Determine the boot/install server's host name by issuing `splstdata -b -l <server_node_number>` and looking at the hostname.

```
# splstdata -b -l 1
                List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response            instal
         last_install_image    last_install_time   next_install_image
------------------------------------------------------------------------
    1 sp2n01             10005AFA18CF    0         disk
            bos.obj.ssp.41  Sat_May__4_10:42:40       bos.obj.ssp.41
```

3. Login to the boot/install server by using `telnet` or the `rsh` command.

   ```
   # telnet sp2n01
   ```

4. Check /etc/bootptab for the node that you are installing. If the node is not listed in this file, then review the NIM client definition as follows:

   - Determine the client's NIM master.

- Determine the NIM master host name.
- Login to NIM master.
- List the objects in the NIM database.
- While still on the NIM master, list the NIM client definition for the node having a problem.
- Compare the values of the *bootp_response* and *Cstate* attributes for the client node.
- If the comparison is ok, then the NIM configuration is correct, and all that needs to be done is to reconfigure boot/install/usr server information for the node in question. If the comparison does not match, then you will have to reconfigure NIM.

5. If the node is listed in this file, continue to Step 2 below:

Step 2: Check Console Messages

1. At the Control Workstation, open a write tty to the node with the install problem by issuing:

   ```
   #s1term -w <frame_number> <node_number>
   ```

2. Wait several minutes to see if any error messages are displayed on the tty that you opened. An error message might tell you what the problem is. Also look for NIM messages on the tty that might suggest that the installation is proceeding. An example of a NIM progress message is:

   ```
   / step_number, of total_steps complete
   ```

Step 3: Check Image Availability

Check to see if the image is available and the permissions are appropriate. Issue:

```
# /usr/lpp/ssp/bin/splstdata -b
```

The `next_install_image` field lists the name of the image to be installed. If the field for this node is set to `default`, the default image (specified by the `install_image` attribute of the SP object) will be installed. The images are found in the /spdata/sys1/install/images directory.

Step 4: Check NIM Configuration for the Node

How to check NIM configuration for a node has already been explained.

Example of resolving Node Install Problem

Here is an example of a problem when installation of node 8, *sp2n08*, failed. Previously the installation process was interrupted for this node. Later Net Boot stopped on LED **231**, which means *attempting a normal mode IPL from Ethernet specified in IPL ROM*. After several retries, the LED code **239** appears, meaning that the boot/install server is not responding. Then an install was tried through *node conditioning*. `Ping` between boot/install server *sp2n01* and client *sp2n08* succeeded, but the install process got stuck on LED **260**, which means displaying information on the display console.

1. `splstdata -b -l 8` pointed to node 1 as the boot/install server for client node 8.
2. `splstdata -b -l 1` provided boot/install server's host name *sp2n01* for boot/install node 1.

3. telnet sp2n01 was enabled to login to the boot/install server node.
4. /etc/bootptab was checked and found that it does not have an entry for *sp2n08*.

   So you now need to check NIM info.
5. Login to *sp2n01* and a check of *lsnim* showed that *sp2n08* is available. lsnim -l sp2n08 listed the following entry for *Cstate*.

   ```
   Cstate        = ready for a NIM operation
   ```

   As the boot response for node 8 was *disk*, *Cstate* value is correct.
6. The following smit command was executed on the Control Workstation:

   ```
   # smitty server_dialog
   ```

   Also, you may use the PSSP command spbootins, as follows:

   ```
   # spbootins -n 1 -r install -h hdisk0 -s yes -l 8
   ```

   The -s yes tells spbootins to run setup_server after changing the SDR values for this node.

   Figure 23 shows the output from setup_server.

```
setup_server command results from sp2n01
setup_server: Starting setup_server

setup_server: Running services_config script to configure SSP services.
This may take a few minutes...

setup_server: Getting Node object information from the SDR

setup_server: Creating Node arrays for processing

setup_server: Getting SP Object information from the SDR

setup_server: Checking to see if this system is an install server

setup_server: Performing node install server setup
setup_server:get_image:no transfer required

setup_server: Checking to see if bos.sysmgt.nim.master is installed
setup_server: bos.sysmgt.nim.master is installed.

setup_server: Checking to see if bos.sysmgt.nim.spot is installed
setup_server: bos.sysmgt.nim.spot is installed.

setup_server: NIM master is configured

setup_server: Checking the NIM master resources - lpp_source, spot, mksys
nst.data, and script
setup_server: spot exists
setup_server: bosinst_data noprompt resource exists
setup_server: bosinst_data prompt resource exists
setup_server: script resource exists

setup_server: Checking NIM client - sp2n08
setup_server: Checking NIM client allocations - sp2n08
setup_server: Allocating resources for client sp2n08
setup_server: Creating the /tftpboot/sp2n08.install_info file
setup_server: Creating the /tftpboot/sp2n08.config_info file
setup_server: Copying /tftpboot/sp2n08-new-srvtab from sp2cw0
```

Figure 23. Output from setup_server

The /etc/bootptab was checked, and now it had an entry for *sp2n08* as follows:

```
sp2n08:bf=/tftpboot/sp2n08:ip=9.12.20.8:
ht=ethernet:ha=10005AFA1B12:sa=9.12.20.1:
sm=255.255.255.0:
```

7. Net Boot was tried again and now it worked.

2.11 Node Customization Problems

When a node is installed, its bootp_response option in the SDR is set to *install*. With this option, the node is network-booted and the installation process begins with the AIX BOS installation. After the first part of the installation is finished, the customization process takes place. This customization process consists of running the pssp_script on the node just installed. This script is the heart of the customization process.

The output of this script is sent to /var/adm/SPlogs/sysman/<node>.config.PID. The variables used by this script are set in <node reliable hostname>.install_info file in /tftpboot on the boot install server at the node being customized or installed. Also, some variables are taken from /sbin/rc.boot file on the node.

The pssp_script installs any additional lpp and device drivers needed on the node, and finally, executes the /tftpboot/script.cust and /tftpboot/tuning.cust scripts after transferring them from the boot install server.

The same script sets back to disk the bootp_response option in the SDR and resets the NIM machine object for that node.

2.11.1 Why a Node Is Not Being Customized

The reasons why a node is not customized are many. However, the reasons will vary, depending on whether the node was being installed, or customized.

- **If This is an Installation**

  Since during the post-install process some lpps are installed along with device drivers needed in the node, the post-install process can fail due to some lpps or device drivers being missing in the lppsource directory. Some times those lpps are deleted to save space, and the problem arises whenever you recreate the SPOT or are trying to install a new node.

  The SPOT must be created with the original lpp sources. For example, if a system was originally installed at AIX 4.1.2 level, and then was later updated to AIX 4.1.4, the AIX 4.1.2 install images must be used as the source to create the /usr SPOT. The resultant /usr SPOT will be at the proper AIX 4.1.4 level when the SPOT creation is complete.

  Alternatively, it is also possible to first create the /usr SPOT at the original level, and then apply all updates to bring the /usr SPOT to the latest level.

  Another problem that can be present during installation or even just customization is the use of short names instead of long names. After applying PTF 12, pssp_script had a problem handling short names. Nodes were stopping with LED u73.

- **If This is a Customization**

The problem is that the perl script mkinstall, called it from setup_server, creates the install_info file using the initial_hostname. However, pssp_script on the other end expects the reliable_hostname.

The mkinstall and mkconfig were changed to use the reliable_hostname instead. Every PSSP component is sensitive to the initial_hostname or reliable hostname variables on the SDR, so keeping them clear will avoid a lot of problems.

Installing the NIM master option in a node that never was a boot/install server was a problem until PTF set 9. The problem was that the installation of this lpp created a /etc/niminfo.prev file that prevented the nodes from being customized.

Anytime you have a problem customizing a node, be sure that pssp_script runs as expected, taking a look at the console log file on the node at /var/adm/SPlogs/sysman/<node>.console.log.

Chapter 3. Kerberos

This chapter briefly describes the role of Kerberos on the RS/6000 SP.

3.1 Overview

The RS/6000 SP currently uses authentication services based on MIT Kerberos version 4. Kerberos functions as a third party to authenticate the identities of clients and servers. Kerberos on the RS/6000 SP is used to initially authenticate the identity of the user and to provide information through which the server can authenticate the identity of clients in a distributed environment. The underlying mechanism for authenticating users and services is a ticket scheme.

For a more detailed explanation of how Kerberos works, see Chapter 14, "Understanding Secure Authentication," in *RS/6000 Scalable POWERparallel Systems: PSSP Version 2 Technical Presentation*, SG24-4542.

3.1.1 Authentication

Authentication refers to the process of checking the correct identity of transmissions; that is, the ability to validate the identity of a user or server.

3.1.2 Authorization

Authorization refers to the process of defining the functions that a user or process is permitted to perform.

Kerberos provides authentication services that allow certain distributed services within the SP system, and between it and other workstations (clients), to securely control access to their services. The root user must use Kerberos (taking on the role of the Kerberos administrator) when installing the SP system, because the installation process includes the creation and modification of the Kerberos security database.

3.1.3 Distributed Commands

Kerberos is also required to use the authenticated distributed commands such as `dsh`, the `p*` commands and `sysctl`.

3.1.4 Remote Commands

Kerberos provides authenticated (Kerberized) versions of `rsh` and `rcp` in the /usr/lpp/ssp/rcmds/bin directory. Customers wishing to use the .rhosts mechanism to restrict access to clients can still do so by using the standard AIX versions of these commands in the /usr/bin directory.

3.1.5 .rhosts

Using Kerberos avoids the need for the root user to have a .rhosts file to control access to network services such as `rsh` and `rcp`.

Kerberos is a method of authentication that is not related to any AIX authentication system, nor is it used as an additional login verification. Therefore, if someone (other than the system administrator) can log into the Control Workstation as root, they can destroy the Kerberos database.

3.2 Terminology

Kerberos clients (user) and services are uniquely identified by a principal identifier, which consists of three components:

- A principal name
- An instance name
- A realm name

3.2.1 Principal

Kerberos defines a name space of authenticated users and services. Each different client and service has a unique *principal* name. An RS/6000 SP user who wishes to use any Kerberos-authenticated service must be registered to Kerberos (by using `kadmin` or `kdb_edit` commands). By virtue of this registration, the user then becomes a Kerberos user (also known as a principal). A private DES key is created for the user and stored in the Kerberos database.

Note that the Kerberos name space is unrelated to the AIX name space, so that an individual may be known by one name to Kerberos and by another name to AIX. However, it is more convenient to assign the same name in each space.

It is possible to have multiple AIX users all using the same Kerberos user to gain access to authenticated services. For example, you can do the following:

1. Define two non-root AIX users, Fred and Joe, on the Control Workstation (make sure that the users are exist on the nodes, as well).
2. Use the command /usr/kerberos/bin/kadmin to add a Kerberos user called kerb.
3. Create a new file, .klogin, in the home directory of the two new AIX users. This file should contain a line similar to kerb@SP21CW0.
4. Log on as either user Fred or Joe and execute `kinit kerb`. The user can now run any of the Kerberos-authenticated commands.

A principal can also refer to a Kerberos-protected service. In this way server programs can be authenticated. For example, the hardmon service principal is used by the hardmon and splogd server daemons.

3.2.2 Instance

The *instance* name is a label that allows the same client or service to exist in several different forms that each require distinct authentication. In the case of services, an instance may specify the host that provides the service. For client principals, the instance can be useful when one wishes to have different identifiers for different privileges. The usual case is that users operate using a name with a null instance.

For example, the client or user principal “root.admin” represents an instance (admin) used for administrative tasks. The service principal “hardmon.sp21cw0” represents an instance (sp21cw0) indicating the node providing the service.

3.2.3 Realm

A *realm* is the set of principals sharing the same authentication database and authentication server. The realm name identifies each independently administered Kerberos site. Kerberos does not specify any constraints on the form of the realm name. When `setup_authent` is run, the realm name is set to the primary authentication server's (usually the Control Workstation) domain name converted to uppercase. If you want to set your own realm name, you must edit the /etc/krb.conf and /etc/krb.realm files.

3.2.4 Ticket

To use a Kerberos service, a client must supply a *ticket* previously obtained from Kerberos. A ticket for a service is a string of bits which have been encrypted using the private key for that service. The ticket contains the following data:

- The name of the client (user)
- The current time
- The length of time the ticket will be valid
- The name of the workstation
- A randomly created DES key (the session key)

3.2.5 Key

The *key* is the password associated with a Kerberos user or service. Keys are stored in the Kerberos database. Keys are used to encrypt the data packets used by Kerberos clients and services.

3.2.6 Ticket-Granting Ticket

When a user executes the `kinit` command, a request is sent to the authentication server. This request contains the user's (principal) name and the name of a special Kerberos service, the ticket-granting service. The authentication server checks if the user is known to Kerberos. If it is, Kerberos creates a random session key and a ticket for the ticket-granting service. This ticket contains:

- The client name
- The name of the ticket-granting server
- The current time
- The lifetime of the ticket
- The client's IP address
- The session key (which was just created)

This ticket is encrypted with a key (using the principal's password) known only to the ticket-granting server and the authentication server and sent back to the client. This ticket is then stored in the user's ticket cache file (/tmp/tkt<uid>) and is known as the ticket-granting ticket. Whenever the client goes back to Kerberos for an additional service-specific ticket, the response is encrypted using the session key from the ticket-granting ticket.

3.3 Components

The following options of the ssp installp image are relevant to Kerberos. The component version numbers included are for PTF 11.

3.3.1 ssp.authent 2.1.0.2

This component contains the authentication server code and the authentication administrator commands. It is installed on the primary authentication server (typically the Control Workstation). This component is not installed on the nodes.

3.3.2 ssp.clients 2.1.0.5

This component is installed on the Control Workstation, the RS/6000 SP nodes, and other RS/6000 hosts where Kerberos is used. It includes:

- All user authentication commands
- kshell services
- spmon command line interface
- Logging daemon

3.4 Install Process

The integration of Kerberos into PSSP scripts is almost transparent to the installation process. The advantage to this is that the RS/6000 SP installations can be completed with a minimal knowledge of Kerberos.

3.4.1 setup_authent

This script creates the primary authentication server. Within an authentication realm, there must be *at least* one authentication server, but you may choose to have more than one. When you configure your realm, you designate one authentication server as the primary server (usually the primary server is the Control Workstation). You may also setup secondary servers. Only the primary server has the kadmind daemon that manages the authentication database.

`setup_authent` does the following:

1. Creates or updates /etc/krb.conf
2. Creates or updates /etc/krb.realms
3. Makes directory /var/adm/SPlogs/kerberos
4. Creates the authentication database using /usr/kerberos/etc/kb_init
5. Creates the master key cache file (/.k) using /usr/kerberos/etc/kstash
6. Adds the kerberos daemon to /etc/inittab and starts it.
7. Adds the kadmind daemon to /etc/inittab and starts it.
8. Defines the initial authentication administrator principal using kdb_ edit
9. Sets up the Kerberos ACLs in /var/kerberos/database/admin_acl.*
10. Runs `kinit` as root.admin and adds the local service principals (hardmon.<CWname> and rcmd<CWname>).
11. Creates /.klogin to authorize the administrator principal to use remote commands.
12. Creates the server key file /etc/krb_srvtab

For further details, see Appendix A, “RS/6000 SP Script Files” on page 229.

3.4.2 install_cw

This creates the hardware monitor ACLs in /spdata/sys1/spmon/hmacls. For further details, see Appendix A, “RS/6000 SP Script Files” on page 229.

3.4.3 setup_server

This creates the Kerberos files intended for the nodes in the /tftpboot directory. For further details, see Appendix A, “RS/6000 SP Script Files” on page 229.

3.4.4 Network Boot

This installs Kerberos on the nodes.

3.5 Daemons and Databases

The following daemons are described in this section:

Kerberos
kadmind
kpropd

The databases used by Kerberos are also described here.

3.5.1 Kerberos Daemon

The kerberos daemon only runs on the primary authentication server (usually the Control Workstation). It is started by the setup_authent script and thereafter invoked from /etc/inittab. This daemon is responsible for providing ticket-granting tickets to clients so that they can access specific server principals.

The kerberos daemon listens for requests on the kerberos4/udp port. If this port is not defined in the /etc/services file, it uses port 750.

Notes:

1. In MIT Kerberos version 5, the kerberos daemon listens for requests on the kerberos5/udp port (port 88).
2. There may be more than one kerberos daemon running on the server. This allows for faster service if there are numerous client requests.

3.5.2 kadmind Daemon

The kadmind daemon only runs on the primary authentication server (usually the Control Workstation). It is started by the setup_authent script and thereafter invoked from /etc/inittab. This daemon is responsible for serving the Kerberos administrative tools, such as changing passwords and adding principals. The kadmind daemon also manages the primary authentication database.

The kadmind daemon listens for requests on the kerberos_master /tcp port. If this port is not defined in the /etc/services file, it uses port 751.

Note: Only one kadmind daemon may run on the server.

3.5.3 kpropd Daemon

The kpropd daemon only runs on secondary authentication servers (if one or more have been set up). The authentication databases used by the secondary authentication servers are copies of the primary database. The databases are maintained by the kpropd daemon, which receives the database content in encrypted form from a program, kprop, which runs on the primary server. Secondary databases can be updated by scheduling execution of a script, /usr/kerberos/ run-kprop, in the root crontab.

The kpropd daemon listens for requests on the krb_prop /tcp port. If this port is not defined in the /etc/services file, it uses port 754.

3.5.4 /etc/services File

The /etc/services file used on PSSP 2.1.0 is not consistent with the previously discussed naming convention for the Kerberos daemon services.

```
kerberos        88/tcp                          # Kerberos
kerberos        88/udp                          # Kerberos
kerberos-adm    749/tcp                         # kerberos administration
kerberos-adm    749/udp                         # kerberos administration
rfile           750/tcp
loadav          750/udp
pump            751/tcp
pump            751/udp
qrh             752/tcp
qrh             752/udp
rrh             753/tcp
rrh             753/udp
tell            754/tcp
tell            754/udp
```

Figure 24. Extract from the /etc/services File

Users must be aware of these potential conflicts when using third party Kerberos servers. If the kerberos and kadmind daemons are running, then the netstat -a command should return the following:

```
root@sp21cw0 / > netstat -a &pipe. grep loadav
udp        0      0  *.loadav               *.*
root@sp21cw0 / > netstat -a &pipe. grep pump
tcp        0      0  *.pump                 *.*                    LISTEN
```

Figure 25. Example Using netstat Command

3.5.5 Databases

The authentication database is created by the command kdb_init, which is called from the setup_authent script. The authentication database is made up of two binary files:

- /var/kerberos/database/principal.pag
- /var/kerberos/database/principal.dir

The Kerberos database contains the name of the authentication realm and all the principals′ names and their keys. The database files can be converted to an ASCII file by the script /usr/lpp/ssp/kerberos/etc/ kdb_util dump. Use kdb_util load to convert the ASCII file back to binary.

3.6 Files

The files described in this section are used by Kerberos.

3.6.1 /.k

The master key cache file contains the DES key derived from the master password. The master password is supplied initially by the administrator when the primary authentication server is created. The corresponding DES key is saved in /.k using the /usr/lpp/ssp/kerberos/etc/kstash command. The kadmind daemon and the database utility commands read the master key from this file instead of prompting for the master password. If the /.k file is deleted, these commands will still execute successfully; however, the user will be prompted for the master password. The kadmind daemon cannot be successfully respawned if the /.k file is removed. The kstash command can be used to recreate the /.k file. The user will, however, be prompted for the master password.

Note: Without a /.k file, the Kerberos server cannot be started automatically during an unattended reboot of the master server.

3.6.2 $HOME/.klogin

The .klogin file contains a list of principals (name.instance@realm). This file specifies the remote principals authorized to invoke commands on the local user account. For example, the root user′s .klogin file contains a list of principals that are authorized to invoke processes as the root user with the Kerberos remote commands (rsh and rcp).

Notes:

1. Only add principals to root′s .klogin file on the Control Workstation.
2. Do not delete any principals which already exist.
3. The root user must always have a .klogin file, and the root user must be listed in the file.
4. The root .klogin is distributed to the nodes during installation or customization. Use dsh to update the nodes′ .klogin file after changing it on the Control Workstation.

3.6.3 /tmp/tkt<uid>

The tkt<uid> file contains the tickets owned by a client (user). The first ticket in the file is the ticket-granting ticket. The ticket cache file is created when the user executes the kinit command. The KRBTKFILE environment variable may be used to change the default location and name for the ticket cache file. The klist command displays the contents of the current cache file. The kdestroy command deletes the current cache file.

3.6.4 /etc/krb-srvtab

The server key file, /etc/krb-srvtab, contains the names and private keys of the local instances of Kerberos-protected services. During the setup of the Control Workstation or the nodes, the keys for service principals are stored in the authenticated database (for use by the authentication server) and in the file /etc/krb-srvtab (for use by the services themselves). So, every node and the Control Workstation includes an /etc/krb-srvtab file that contains the keys for the services provided on that host. On the Control Workstation, the hardmon and rcmd service principals are in this file:

```
root@sp21cw0 / > klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
hardmon         sp21tr0         SP21CW0         1
rcmd            sp21tr0         SP21CW0         1
hardmon         sp21cw0         SP21CW0         1
rcmd            sp21cw0         SP21CW0         1
```

Figure 26. Example of /etc/krb-srvtab from the Control Workstation

On the nodes, the rcmd service principals are in this file:

```
root@sp21n01 / > klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
rcmd            sp21n01         SP21CW0         1
```

Figure 27. Example of /etc/krb-srvtab from a Node

Note: Always ensure that the service keys contained in the authentication database and in the /etc/krb-srvtab files on the nodes match. The /usr/lpp/ssp/kerberos/etc/ext_srvtab command can be used to create new server key files for each node.

3.6.5 /etc/krb.conf

The SP authentication configuration file, /etc/krb.conf, defines the local authentication realm and the location of authentication servers for known realms. The first line contains the name of the local authentication realm. Subsequent lines specify the authentication server for a realm.

This file is created by the setup_authent script on the primary authentication server. You may supply your own krb.conf file before running setup_authent if you want to use a non-default realm name (the default realm name is the domain portion of the primary authentication server's hostname converted to uppercase).

```
root@sp21n01 / > cat /etc/krb.conf
SP21CW0
SP21CW0 sp21cw0 admin server
```

Figure 28. Example of a /etc/krb.conf File

3.6.6 /etc/krb.realms

This file maps a host name to an authentication realm for the services provided by that host. Each line in the file must be in one of the following forms:

- host_name realm_name
- domain_name realm_name (Domain names should begin with a period)

```
sp21tr0 SP21CW0
sp21n01 SP21CW0
sp21n02 SP21CW0
sp21n03 SP21CW0
sp21n04 SP21CW0
sp21n05 SP21CW0
sp21n06 SP21CW0
sp21n07 SP21CW0
sp21n08 SP21CW0
sp21n09 SP21CW0
sp21n10 SP21CW0
sp21n11 SP21CW0
sp21n13 SP21CW0
sp21n15 SP21CW0
sp21sw01 SP21CW0
sp21sw02 SP21CW0
sp21sw03 SP21CW0
sp21sw04 SP21CW0
sp21sw05 SP21CW0
sp21sw06 SP21CW0
sp21sw07 SP21CW0
sp21sw08 SP21CW0
sp21sw09 SP21CW0
sp21sw11 SP21CW0
```

Figure 29. Example of a /etc/krb.realms File

This file is created (if it does not yet exist) by the setup_authent script on the primary authentication server.

3.6.7 /var/adm/SPlogs/kerberos

There are two log files in the /var/adm/SPlogs/kerberos directory:

- kerberos.log
- admin_server.syslog

The kerberos.log file contains information relating to the kerberos daemon, such as when it was started. It also contains error messages. Following is an example of the kerberos.log file containing the error messages when the /.k file is deleted.

```
26-Apr-96 14:48:04 Kerberos started, PID=17802
26-Apr-96 14:48:25 Kerberos started, PID=17818
26-Apr-96 14:48:25 kerberos: 2503-000 Could not read master key.
26-Apr-96 14:48:25 Kerberos will pause so as not to loop init
26-Apr-96 14:59:21 Kerberos started, PID=41620
26-Apr-96 15:19:11 Kerberos started, PID=57046
```

Figure 30. Example of kerberos.log File

The admin_server.syslog file contains information relating to the kadmind daemon. It also contains error messages from the daemon. Following is an example of the admin_server.syslog file containing the error messages when the /.k file is deleted.

```
26-Apr-96 14:49:23 Kerberos admin server started, PID=57080
26-Apr-96 14:49:23 kadmind: 2503-101 error: 2504-317 Could not fetch master key
26-Apr-96 14:49:23 Shutting down admin server
26-Apr-96 14:53:17 Kerberos admin server started, PID=22380
```

Figure 31. Example of admin_server.syslog File

3.7 Commands

This section describes the commands used by Kerberos.

3.7.1 kinit

The `kinit` command is used to authenticate a user to the Kerberos authentication services. A ticket-granting ticket is created and stored in the user's ticket cache file (usually /tmp/tkt<uid>). All the user's previous tickets are destroyed.

The simplest way to invoke the `kinit` command is to specify the principal as the only argument; for example:

```
kinit fred
kinit root.admin
```

3.7.2 klist

The `klist` command displays the principal name, the issuing time, and the expiration time for each ticket held in the user's current cache file. The ticket-granting ticket is displayed first, followed by the various service tickets.

Following is an example of the output of the `klist` command.

```
root@sp21cw0 / > klist
Ticket file:    /tmp/tkt0
Principal:      root.admin@SP21CW0

  Issued           Expires          Principal
Apr 26 14:59:45  May 26 14:59:45  krbtgt.SP21CW0@SP21CW0
Apr 26 15:00:10  May 26 15:00:10  hardmon.sp21cw0@SP21CW0
Apr 26 15:38:50  May 26 15:38:50  rcmd.sp21n01@SP21CW0
Apr 26 15:38:50  May 26 15:38:50  rcmd.sp21n02@SP21CW0
Apr 26 15:38:50  May 26 15:38:50  rcmd.sp21n03@SP21CW0
```

Figure 32. Example of klist Output

The output lists the location of the principal's current cache file, as well as the principal's full name. The first ticket is the ticketgranting ticket. The service tickets are for the System Monitor service (hardmon) and for the SP Remote Command service (rcmd) on nodes 1, 2, and 3.

The command `klist -srvtab` displays the local instances of services and their private key versions found in the server key table (usually /etc/krb-srvtab).

Following is an example of the output of the `klist -srvtab` command.

```
root@sp21cw0 / > klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
hardmon         sp21cw0         SP21CW0         1
rcmd            sp21cw0         SP21CW0         1
rcmd            sp21tr0         SP21CW0         1
hardmon         sp21tr0         SP21CW0         1
```

Figure 33. Example of klist -srvtab Output

The output displays the key versions for service principals on the Control Workstation.

3.7.3 kdestroy

The kdestroy command writes zeros to the user's current ticket cache file and then removes the file. As a result, all the user's active authentication tickets are deleted. For added security, you may wish to automatically destroy your tickets when you log out.

3.7.4 kstash

The kstash command saves the system's authentication database master key in the master key cache file (/.k). The user is prompted to enter the master key to verify the authenticity of the key and to authorize caching it.

3.7.5 dsh and p* Commands

The dsh command uses rsh to execute a specific AIX command on any group of nodes or other remote RS/6000 hosts within the authentication realm, in parallel. The group of target hosts may be pointed to by the WCOLL variable, which in turn points to a file containing the hostname of each target host.

There are various p* commands (p_cat, pcp, pdf, pfck), which all use dsh to execute a specific AIX command in parallel on multiple hosts.

3.7.6 rsh and rcp

There is a Kerberos-authenticated version of both rsh and rcp in the /usr/lpp/ssp/rcmd/bin directory. To use the Kerberos versions, the user must include this directory before the /usr/bin directory in the local host's PATH.

If the user's authentication fails, the /usr/lpp/ssp/rcmd/rsh (or rcp) command issues an error message and passes its arguments to /usr/bin/rsh (or rcp). In this case the user will require normal remote command access to the remote host (through /etc/hosts.equiv or $HOME/.rhosts).

3.7.7 sysctl

The sysctl command provides a command line interface for communicating with the sysctl remote command execution and monitoring server, sysctld. Sysctl connects to a remote host's sysctld using TCP/IP, passes keywords and commands to the server, and writes output returned to stdout. Any sysctl user must be a Kerberos principal.

3.8 Solving Problems

There are a few basic things to check when Kerberos authentication fails. The following hints and tips should be followed before taking the more drastic measure of rebuilding the entire Kerberos database. Also consult Chapter 6, "Diagnosing Authentication Problems," in the *RS/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*, GC23-3899 for further information.

3.8.1 Daemons

Start by checking that both the kerberos and the kadmind daemons are running. If the daemons are not running, then check /etc/inittab. The entries for these two daemons should be similar to the following:

```
root@sp21cw0 /etc > lsitab kerb
kerb:2:respawn:/usr/lpp/ssp/kerberos/etc/kerberos
root@sp21cw0 /etc > lsitab kadm
kadm:2:respawn:/usr/lpp/ssp/kerberos/etc/kadmind -n
```

Figure 34. inittab Entries for Kerberos Daemons

If the /etc/inittab file is correct, then try to start the daemons from the command line. Also check the Kerberos daemon log files in the /var/adm/SPlogs/kerberos directory for any error messages, which may explain why the daemons cannot be respawned.

3.8.2 Tickets

Confirm that the user is known to Kerberos and that the user has a valid ticket for the requested service by runnig the `klist` command. Check that the user's ticket cache file exists. If the ticket cache file is not in the default location (/tmp/tkt<uid>), then see where the KRBTKFILE variable points. If the problem appears to be due to invalid tickets, then use the `kdestroy` and `kinit` commands to generate new tickets for the user.

Also check that the $HOME/.klogin file exists and that there is an entry for the user (requesting the service) in the file.

3.8.3 PATH Variable

Ensure that the /usr/lpp/ssp/rcmd/bin directory is before the /usr/bin directory in the user's local PATH statement. Also make sure that the PATH variable is exported.

```
PATH=/usr/lpp/ssp/rcmd/bin:$PATH:/usr/lpp/ssp/bin
 :/usr/lpp/ssp/kerberos/bin:/usr/local
export PATH
```

Figure 35. Example of PATH Statement from .profile File

3.8.4 Configuration Files

Check and confirm that the contents of the /etc/krb.conf and /etc/krb.realm files are correct.

```
SP21CW0
SP21CW0 sp21cw0 admin server
```

Figure 36. Sample /etc/krb.conf File

```
sp21tr0 SP21CW0
sp21n01 SP21CW0
sp21n02 SP21CW0
sp21n03 SP21CW0
sp21n04 SP21CW0
sp21sw01 SP21CW0
sp21sw02 SP21CW0
sp21sw03 SP21CW0
sp21sw04 SP21CW0
```

Figure 37. Sample /etc/krb.realms File

3.8.5 TCP/IP

Check that TCP/IP is functioning correctly by using the ping command to confirm communication with the various adapters on each node. Also use the host <hostname> and host <IP address> commands to ensure that hostname resolution is correct (both commands must return the same output).

```
root@sp21cw0 /tmp > host sp21cw0
sp21cw0 is 9.12.60.99,  Aliases:  sp21en0

root@sp21cw0 /tmp > host 9.12.60.99
sp21cw0 is 9.12.60.99,  Aliases:  sp21en0
```

Figure 38. Example of Host Name Resolution

3.8.6 Remote Principals

Ensure that the remote user or client has permission to invoke commands on the local user account. If the originating remote user is authenticated to one of the principals named in the .klogin file, access is granted to that account.

Note: If a klogin file is present, the owner must also be listed in order to gain access to the user's own account from a remote host.

3.8.7 Service Key Files

First check that the service key files (/etc/krb-srvtab) exist on all nodes. Use either the ksrvutil or the klist command to show the version numbers of the service keys in the /etc/krb-srvtab files.

```
root@sp21cw0 /etc > klist -srvtab
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
-----------------------------------------------------
hardmon         sp21cw0         SP21CW0         1
rcmd            sp21cw0         SP21CW0         1
rcmd            sp21tr0         SP21CW0         1
hardmon         sp21tr0         SP21CW0         1
```

Figure 39. Example of klist -srvtab Command on the Control Workstation

The service key version numbers in the /etc/krb-srvtab files must match the version number kept in the Kerberos authentication database. Use the /usr/lpp/ssp/kerberos/etc/kdb_util command to dump the contents of the authentication database to an ASCII file. The fifth field in this file is the service key version number.

If there are any problems with the service key files, then they must be re-created using the /usr/lpp/ssp/kerberos/etc/ext_srvtab command. See Chapter 6, "Diagnosing Authentication Problems," in the *RS/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*, GC23-3899 for further information.

3.8.8 PTF Levels

Ensure that the Control Workstation and all the nodes are at the same PTF level.

3.8.9 Rebuild the Kerberos Database

If all else fails, the Kerberos authentication database may be completely re-created. The following steps may be used to achieve this.

Note: This procedure does not require the user to reboot the nodes.

1. Ensure the following directories are included in your PATH:
   - /usr/lpp/ssp/kerberos/etc
   - /usr/lpp/ssp/kerberos/bin
   - /usr/lpp/ssp/bin
2. kdb_destroy (delete the kerberos database)
3. kdestroy (delete all current tickets)
4. rm /.k (delete the master key cache file)
5. rm $HOME/.klogin
6. rm /etc/krb* (delete the Kerberos configuration files)
7. chitab "kadm:2:off:/usr/lpp/spp/kerberos/etc/kadmind -n"
   chitab "kerb:2:off:/usr/lpp/spp/kerberos/etc/kerberos"
8. telint 2 (refresh inittab)
9. stopsrc -s hardmon
10. setup_authent
11. spbootins -r customize -l $NODELIST
    (where $NODELIST is a comma-separated list of node numbers)
12. startsrc -s hardmon
13. telinit 2 - Restart the Kerberos daemons
14. ext_srvtab -n $NODENAMES
    (where $NODENAMES is a comma-separated list of node names). This creates files called <nodename>-new-srvtab in the current directory). You should run this command against all interfaces, then you should concatenate those file before transfer them to the nodes.
15. ftp the files created in Step 14 to their respective nodes. Place the files in /etc and name them krb-srvtab.
16. Check that /etc/krb.conf has the entry rcmd.CWname.Realm
    (where Realm is the realm specified in /etc/krb.realms).
17. Check /etc/krb.conf and /etc/krb.realms on all nodes.
    (These files should be the same as the ones on the Control Workstation.)
18. From the CW: dsh -a id (Check that kerberos works.)
19. From the node1: dsh -w node2 id
20. spbootins -r disk -l $NODELIST

Figure 40. How to Rebuild the Kerberos Database

Chapter 4. The Switch

Most of the switch-related problems require a clear understanding of the switch components. This chapter explains those components, gives examples about configuration and topology files, and explains how to handle switch problems and interpret switch log files.

4.1 Overview

In the following section, the term *switch* is used generically to cover both the High Performance Switch and the new SP Switch. Where the comments apply to only one of these switches, the appropriate name will be used.

The assumption is also made that the switch board described is *not* the LC8 Switch board which can be installed in the half height frames (49 inch frame); this only provides connections to 8 nodes. This switch board will be discussed in detail later on.

The switch consists of 3 main hardware components:

- The switch board containing the switch chips
- The switch cables to connect the switch to the nodes and other switches
- The switch adapters for each node that is to be part of the switch network

Each switch board has 32 sockets available to connect to either nodes, or other switches (that will be part of the same network). Sixteen sockets are available for the nodes, and 16 sockets are available for the external communications to the other switches.

These socket connectors are known as *Jack Sockets*, which are abbreviated in much of the documentation to *Jxx* (where *xx* is the number that identifies a particular socket).

J3 through to J34 are available for these types of communications. J1 and J2 are used for alternative communications, such as the serial connection. The power source is the other usage of these connections.

Figure 41 on page 88 shows an example of these connections for the High Performance Switch.

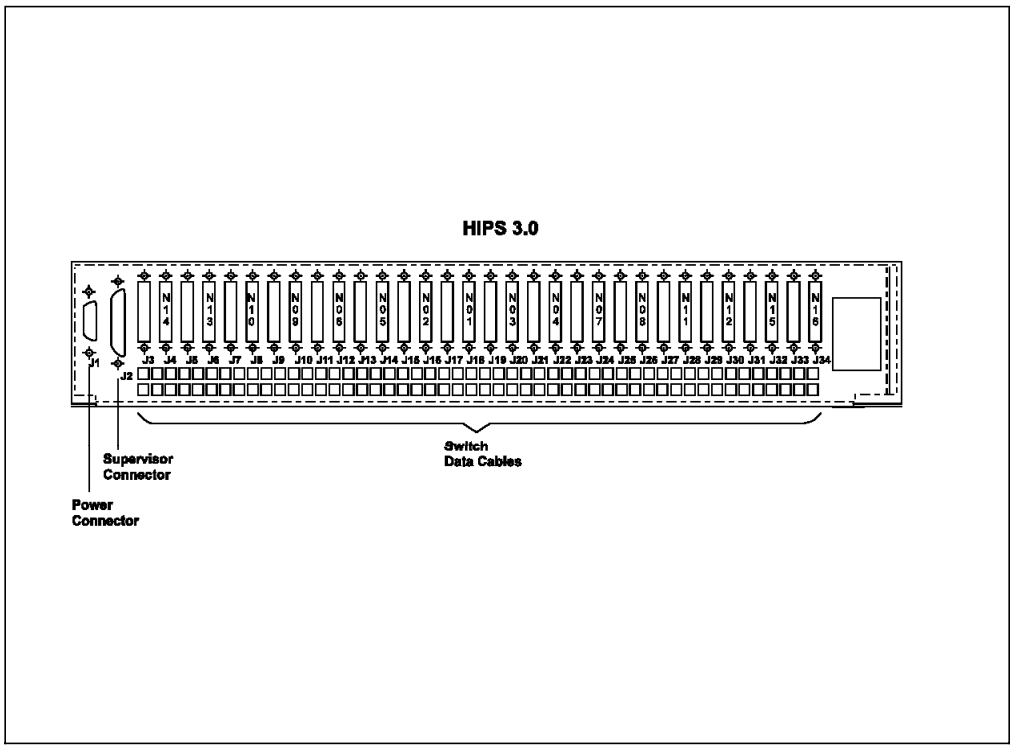


Figure 41. The HiPS Showing the External Connections

It is worth stating at this point that the Control Workstation is not part of the switch network, and putting a switch adapter in the Control Workstation is not supported in the current environment.

Setting aside the LC8 Switch for the time being, there are 2 main types of switch boards that can be used:

- Node switch boards
- Intermediate switch boards

To explain why intermediate switch boards are required, it is necessary to return to the comment made earlier that 16 *jack connections* are available for communication with other switches. If you take the situation where there are 2 switches in the network, all 16 of these are available to connect to the other switch (despite the fact that data will only flow down 8 of these). If there are 3 switches, then 8 cables will go to the first switch and 8 to the other.

The number of cables going to each switch decreases to the point where you have 5 switches allowing only 4 cables to go to each switch. Having more than 5 switches in a configuration is not currently supported (although there is no intrinsic reason why it cannot be done) because by having less than 4 cables to connect each switch, the bandwidth of the network starts to be compromised.

The intermediate switch boards remedy this situation. Special frames that contain only switches and their associated cables are required to house these boards. These switch boards only communicate with other switches for which there are all 32 jack connections available. Within these switch frames there are always 4 intermediate switch boards, all of which are cabled to the node switch boards for configurations up to the 128-way size. The 4 switch boards, with 32 connections each, make 128 connections possible.

Actually, for 128-way systems, while all 16 available connections on the node switch boards are cabled to the intermediate switch boards (4 to each of the ISBs), only half of these cables are actually used to transfer data. It is by taking out these *redundant* cables and using the connection on this ISB to connect to other ISBs that it is possible to go beyond the 128-way to larger switch sizes. The jack socket on the NSB is covered with a blanking plate in this instance.

Figure 42 shows how ISBs are cabled for two of the four ISBs (the diagram would be rather confusing if all 4 were included). In addition, there is reference to connections between ISB frames. These are only required for systems larger than the 128-way.

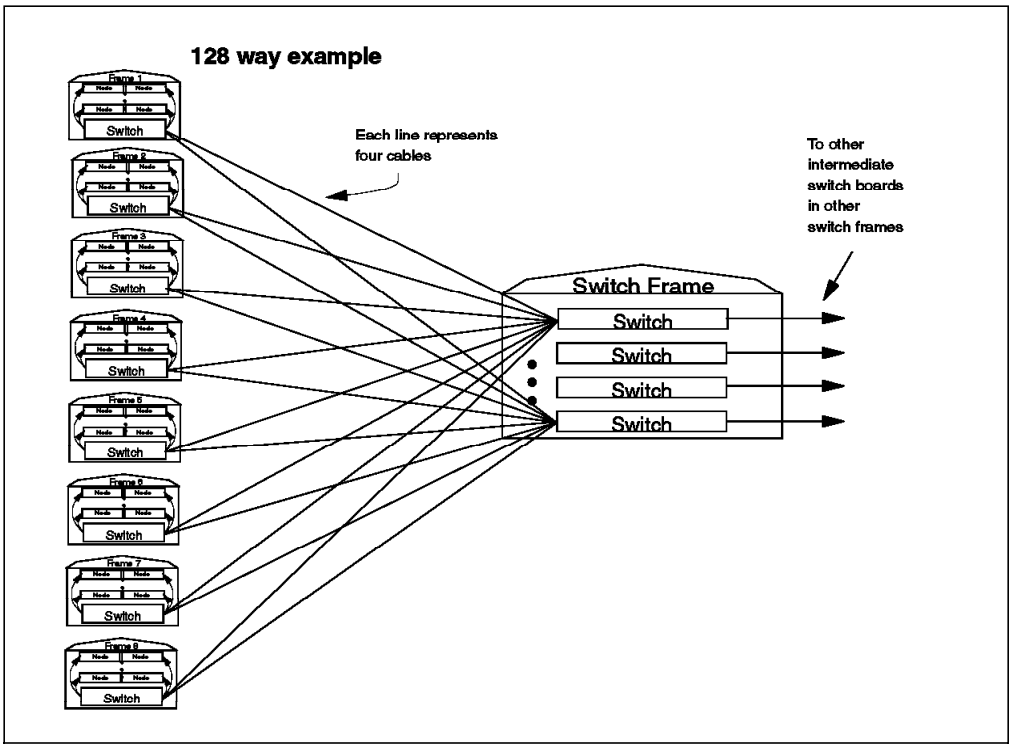


Figure 42. Example of the Cabling for a 128-Way System

The connectivity on 128-way systems is excellent because there are always 8 cables that data can travel down to reach the node frame. The intermediate switch boards provide sufficient routes to ensure that this is an optimum configuration. If a 128-way configuration was supported without the use of intermediate switch boards, then only 2 cables could be used to connect the majority of the switch boards and the bandwidth would be compromised.

When using the intermediate switch boards, there will be more hops required to travel between the node switch boards because the route must always go through one of these intermediate boards rather than directly from one node switch board to another. This increases the latency of the network, but because the latency is so low on the switch, this does not become a significant factor in the speed of the network. The determining factor is whether there is contention for routes. As soon as packets are delayed due to all available routes being busy, then network delays occur. By using intermediate switch boards, packet contention is kept to a minimum. In the hypothetical example given for a 128-way system with all switch boards directly attached, packet contention is likely to occur on a more regular basis.

Figure 43 on page 90 gives an example of the cabling for a 3-switch system without using intermediate switch boards.

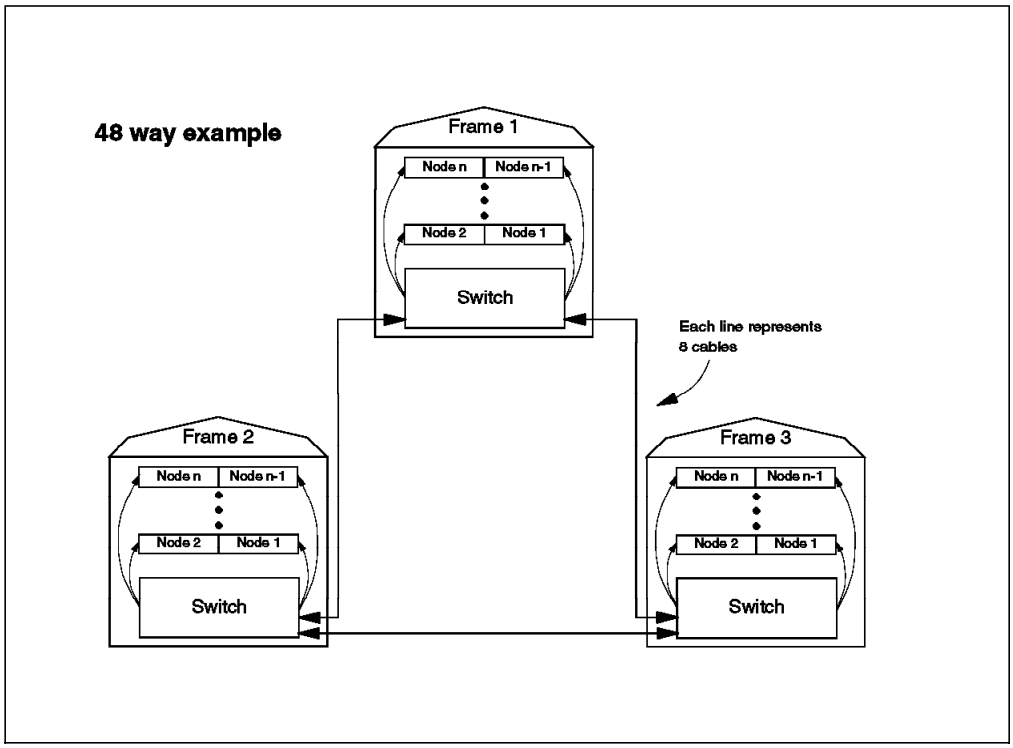


Figure 43. Example of the Cabling on a 48-Way System

The exact cabling configuration will be addressed later (that is, which jack connection goes to which node or other jack connectors on other switches), but first the internal cabling on the switch board will be covered.

4.1.1 Software Overview

The High Performance Switch can be run at both PSSP 1.2 and PSSP 2.1. The SP Switch, though, can only be run at PSSP 2.1 and only with a certain level of maintenance applied. To check your current level of maintenance, execute:

```
# lslpp -l | grep ssp
```

If you have to install the SP Switch, ensure that the following two products are at least at the levels specified below:

```
ssp.basic   2.1.0.10      SP System Support Package
ssp.css     2.1.0.8       SP Communication Subsystem
```

This level of code is known as *PTF*set11. In fact, the levels should be higher than this to avoid the known problems that can be associated with newly-supported devices.

If the High Performance Switch is installed, the level of maintenance shown above (ssp.basic 2.1.0.10 and ssp.css 2.1.0.8) is perfectly valid. It provides support for both types of switches.

You may also use CSS_test command to verify that the same PSSP level is installed on the nodes and on the Control Workstation.

4.1.2 High Performance Switch and SP Switch Coexistence

Note

In the current environment, it is not possible for the High Performance Switch and the SP Switch to coexist within the same RS/6000 SP system. If there is a requirement to install a new SP Switch without migrating the existing High Performance Switch, then a new system, which has its own Control Workstation and its own SP environment, is required.

The cabling is slightly different on the SP Switch. The jack connectors are now associated with different nodes and different jack connectors on other switches. For example, Node 14 (N14) is connected to the switch board at J4 on the High Performance Switch, but on the SP Switch it is connected at J34.

The topology files which describe the cabling are still the same.

Systems with the SP Switch installed do not support system partitions with nodes that have anything less than the required software levels specified earlier (that is, PSSP 2.1 with PTF set 11). The support is provided at this level, and so AIX Version 3.2.5 partitions are not possible in this environment.

From the application perspective, no changes should be required when moving to the SP Switch environment. Functionally, the switch looks the same to the application. For example, it has the same topology, and the same structure.

4.1.3 Comparing HiPS and SP Switch

- Both switches are packet switches with wormhole routing.

  Packet switches with wormhole routing help balance low latency and help to relieve of congestion problems. Circuit switches open the entire path from sending node to receiving node and keep that path through a whole message. This reduces latency and guarantees delivery, but it also locks up resources and can cause heavy congestion. Packet switching only locks up a resource for the time of a packet. Using wormhole routing reduces latency because it allows you to start sending a packet as soon as a receive buffer is available. However, when a packet gets blocked downstream, it can back up through several switch chips and start to have congestion problems. So, by using packet switching with wormhole routing, latencies are reduced.

- The board is still 16x16x2.

  There are 16 ports on one side of the board that are connected to four chips. These four chips are connected to four chips on the other side of the board, which are in turn connected to 16 ports that also leave the switch assembly. So, now you can picture the 16x16. The “2” comes from the ports being duplex; data passes in both directions. Figure 42 on page 89 and Figure 43 on page 90 show examples.

- Global synchronous clocks.

  Global synchronous clocks allow you to easily keep the same Time of Day (TOD) across the system. This helps applications and fault isolation. If a particular error is propagated through a system (for example, if there is a bad Error Detection Code (EDC) on some data), you can now see it first by comparing the time stamp on various switch chips′ error register. This will allow you to determine where the problems first occurred.

The problem that you have with global synchronous clocks is that if the master fails, the entire system goes down. SP Switch has improved recoverability by putting two oscillators on each board. However, the switchover will probably still be a manual process for a while. This is because switching to a new oscillator will bring down the system. If the hardware and software incorrectly sense that there is a clock problem, this could bring down the whole system for no good reason. Until we have a better handle on how to always correctly determine that a clock failure has occurred and we can always correctly determine how to recover, we will leave this up to a human operator who has a better understanding of the state of the user applications when the system is brought down.

- They use the same topology files

  The topologies in HiPS are extremely scalable, so the SP Switch uses the same files. *Eannotator* must be run over the topology files to update the new wiring. Refer to 4.5, “Switch Commands” on page 104.

- They have the same mechanical factor

4.1.4 Improvements of SP Switch over HiPS

- Faults are not global

  The HiPS had global faults. When a fault occurred, a fault code was propagated throughout the entire switch network until all of the links were brought down and had to be reinitialized. This is a remnant of the original design point of the HiPS switch as it was designed in research. The original machine was only supposed to be running a single application.

  With SP Switch, we had time to redesign the fault scenarios and we made the faults localized. Only the link that experienced the fault is brought down. It can then be brought back online with minimal disturbance to the system. You will be glad to know that taking a node offline will no longer bring down the entire switch.

- No more chip shadowing

  The HiPS used two physical chips to perform the function and checking of one logical chip. The second chip checked the operation of the first chip. This is a very thorough way to detect errors. However, it is also very expensive. You have to add some circuitry to check the other chip and you use two chips. By adding a little more circuitry to one chip, you can do checking that is almost as good, but a lot cheaper. The architecture of the system (with EDC and CRC on the messages) also helps check messages for errors. What is lost is a little fault isolation. For that we get a less expensive design with better reliability (fewer parts to break).

- Cable clips on; not screwed in

  An annoying problem with HiPS was that each of the 32 cable/wrap plugs on a switch had to be screwed in: 2 screws on 32 cable is 64 screws. This slowed down installing and servicing the switch. It also caused some sore wrists for people installing large systems. SP Switch addressed this by putting clips on the connectors instead of screws. Now, clips are not perfect. They are not seated as surely as screws are. You also will have a small pitch between connectors, so getting your fingers in to work the clips is not necessarily a simple task.

- Estimated 12.8 times better reliability

With newer technology, the SP Switch is more reliable that HiPS. Also, fewer parts improves the reliability. The function of the HiPS 32 driver/receiver chips embedded in the SP Switch chips. This is also true of most of the clocking logic. The SP Switch board is basically eight switch chips, two oscillators and four phase-locked loop (PLL) chips. Of course, you still have terminating resistors and decoupling capacitors.

- N+1 power supplies and fans

  The N+1 power supplies and fans will help availability and allow you to defer maintenance on failed components to a more convenient time.

- 150 MBps single direction bandwidth

  The switch provides 150 MBps single direction bandwidth (300 MBps both directions). This is nominal bandwidth. The effective bandwidth will change according to the different protocol stacks used to transfer data.

- The link is more advanced

  Internal and external links have a more advanced technology that provides greater speed, and that can also adjust itself better to drifts in the clock/data relationship that are caused by things like temperature changes and component variations caused by age.

4.2 Reviewing Switch Boards

The following sections reviews switch board information, as illustrated in Figure 44.

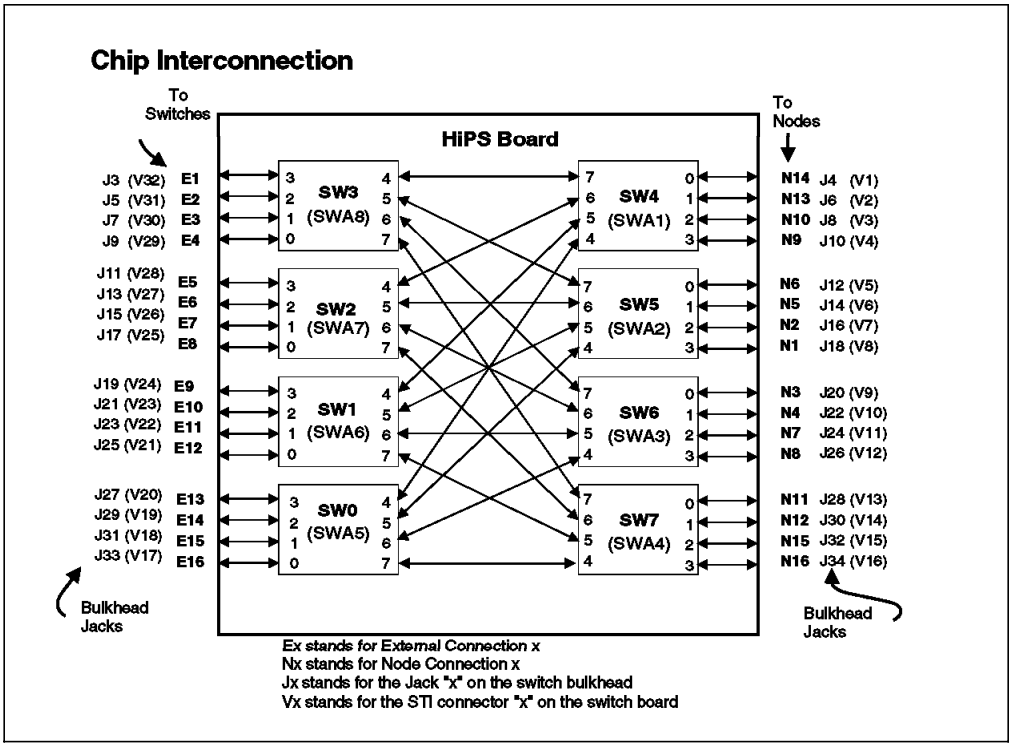


Figure 44. The HiPS Board

J3 to J34 Bulkhead jacks for data cables

V1 to V32 Data connectors on the switch board

SW0 to SW7 Logical identification of switch chips

- In output files, add 10,000 to the logical number.
- The frame number is used to build the logical number - multiply frame number by 10.

SWA1 to SWA8 Master switch chip silkscreen on the board

N1 to N16 Node ports (physical numbering)

E1 to E16 Switch to switch ports

Each switch board contains eight logical switch chips. In fact, on the High Performance Switch, there are two physical chips per logical switch chip. Each chip has eight ports to communicate with the nodes, other switch chips, or other switches.

The switch chips numbered SW4, SW5, SW6 and SW7 handle all the communications with the nodes using chip ports 0, 1, 2 and 3. The other four chip ports (4, 5, 6 and 7) are used to communicate with the other switch chips: SW0, SW1, SW2 and SW3. By routing across to these four chips on the other side, it is possible for each of the chips that service the nodes to reach the other three node chips through four different routes.

The following example illustrates this concept.

To communicate from node 14 (N14) to node 16 (N16), the path that a packet could take is as follows:

- The packet passes SW4 through port 0, and can exit the switch chip through the following four routes, which are the chosen routes:
  1. Port 7 across to SW3, onto that chip through port 4, exiting on port 7, over to SW7, onto the chip on port 7
  2. Port 6 across to SW2, onto that chip through port 4, exiting on port 7, over to SW7, onto the chip on port 6
  3. Port 5 across to SW1, onto that chip through port 4, exiting on port 7, over to SW7, onto the chip on port 5
  4. Port 4 across to SW0, onto that chip through port 4, exiting on port 7, over to SW7, onto the chip on port 4
- Once onto SW7, it will exit through port 3, go through J34 onto the cable, and onto the switch adapter on node 16 (N16).

This mechanism ensures that there are four routes between each node on a single switch board in a single partition system.

The switch chips SW3, SW2, SW1, and SW0 handle the communications to the other switch chips, as well as providing the four routes between all the nodes as just described. These chips handle the communications to the other switch boards on ports 0, 1, 2, and 3 (which are unused on a single switch system) and using ports 4, 5, 6, and 7 to communicate with the chips that connect directly to the nodes (SW4, SW5, SW6, and SW7).

The connections to other switches will vary, depending on how many switches are being connected. This will be dealt with later after the switch topology files have been discussed.

The cabling is slightly different on the SP Switch. The jack connectors are now associated with different nodes and different jack connectors on other switches.

For example, Node 14 (N14) is connected to the switch board at J4 on the High Performance Switch, but on the SP Switch it is connected at J34. Look at Figure 45 on page 95 and compare it with Figure 41 on page 88 for the High Performance Switch to discover how it is cabled differently.

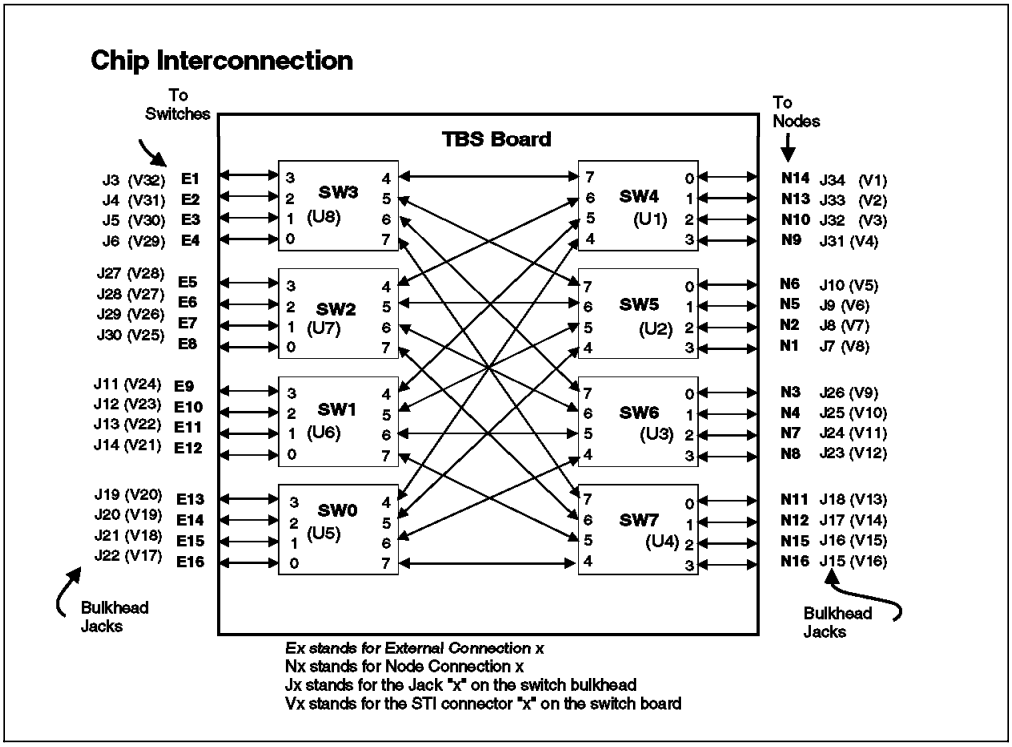


Figure 45. The SP Switch Board

J3 to J34 Bulkhead jacks for data cables

V1 to V32 Data connectors on the switch board (silk screen)

SW0 to SW7 Logical identification of switch chips

- In output files, add 10,000 to the logical number.
- The frame number is used to build logical number. Multiply frame number by 10.

U1 to U8 Master switch chip silkscreen on the board

N1 to N16 Node ports (physical numbering)

E1 to E16 Switch to switch ports

4.3 Switch Topology and Clock Subsystem

The following section discusses switch topology and clock subsystem considerations.

4.3.1 Switch Topology Files

The switch topology file is critical to the successful working of the switch. The switch daemons (which will be discussed in some detail later) use this topology file as the basis for creating the routing tables which get propagated to every node on the switch network. The topology files are shipped as standard with the POWERparallel System Support Programs software, and during the installation

process, you are required to choose the appropriate one, based on the number and type of switch boards that you have installed.

The file expected.top.*NSBnum*nsb.*ISBnum*isb.*type* located on the Control Workstation describes the wiring configuration for the switch. The switch topology file naming convention is as follows:

expected.top.*NSBnum*nsb.*ISBnum*isb.*type*

Where:

NSBnum (NodeSwitchBoard Number) is the number of NSBs in the configuration

ISBnum (Intermediate SwitchBoard Number) is the number of ISBs in the configuration

type This is the type of topology. The default is 0.

Note: This file must not be changed; otherwise the configuration is not supported. All RS/6000 SP systems are cabled in a standard way that follows the wiring set out in the topology files. This pattern of cabling or the contents of the topology files should only be changed for diagnostic purposes on the advice of an IBM engineer.

NSBs are switches mounted in frames containing nodes. ISBs are switches mounted in switch expansion frames. ISBs are used in large systems to connect many processor frames together so that bandwidth is not compromised on the switch network.

Following is a listing from the /etc/SP directory on the Control Workstation. While the Eclock files reside in this directory at PSSP 2.1, the topology files are merely symbolic links to the directory:

```
/spdata/sys1/syspar_configs/topologies
```

In PSSP 1.2, these files did reside in the /etc/SP, but the change was made to make the directory structure consistent with System Partitioning that was introduced in PSSP 2.1.

```
[root@sp21cw0] /etc/SP > ls
Eclock.top.1nsb.0isb.0      expected.top.1nsb.0isb.0
Eclock.top.1nsb_8.0isb.0    expected.top.1nsb_8.0isb.0
Eclock.top.2nsb.0isb.0      expected.top.1nsb_8.0isb.1
Eclock.top.3nsb.0isb.0      expected.top.2nsb.0isb.0
Eclock.top.4nsb.0isb.0      expected.top.3nsb.0isb.0
Eclock.top.4nsb.2isb.0      expected.top.4nsb.0isb.0
Eclock.top.5nsb.0isb.0      expected.top.5nsb.0isb.0
Eclock.top.5nsb.4isb.0      expected.top.5nsb.4isb.0
Eclock.top.6nsb.4isb.0      expected.top.6nsb.4isb.0
Eclock.top.7nsb.4isb.0      expected.top.7nsb.4isb.0
Eclock.top.8nsb.4isb.0      expected.top.8nsb.4isb.0
```

The active topology file for a system partition contains within it the data required by CSS switch code to maintain the switch fabrics. In particular, it contains representations of:

- All (possible) available frames of the system
- All (possible) available nodes of the system
- All (possible) available switch boards of the system

- The board, switch chip and port at which each node is attached
- The chip-to-chip connections within each board, including port data
- The interboard connections, including chip and port data

The contents of a given topology file are enhanced by running Eannotator against it. This adds comments to the records of the file, namely jack information which help the reader and CSS switch code in understanding the cabling between elements of the system. Eannotator embodies an understanding of the cabling used in the system of a given switch type.

```
format 1
16 18
# Node connections in frame LO1 to switch 1 in LO1
s 15 3  tb0 0 0     LO1-S00-BH-J18 to LO1-N1
s 15 2  tb0 1 0     LO1-S00-BH-J16 to LO1-N2
s 16 0  tb0 2 0     LO1-S00-BH-J20 to LO1-N3
s 16 1  tb0 3 0     LO1-S00-BH-J22 to LO1-N4
s 15 1  tb0 4 0     LO1-S00-BH-J14 to LO1-N5
s 15 0  tb0 5 0     LO1-S00-BH-J12 to LO1-N6
s 16 2  tb0 6 0     LO1-S00-BH-J24 to LO1-N7
s 16 3  tb0 7 0     LO1-S00-BH-J26 to LO1-N8
s 14 3  tb0 8 0     LO1-S00-BH-J10 to LO1-N9
s 14 2  tb0 9 0     LO1-S00-BH-J8  to LO1-N10
s 17 0  tb0 10 0    LO1-S00-BH-J28 to LO1-N11
s 17 1  tb0 11 0    LO1-S00-BH-J30 to LO1-N12
s 14 1  tb0 12 0    LO1-S00-BH-J6  to LO1-N13
s 14 0  tb0 13 0    LO1-S00-BH-J4  to LO1-N14
s 17 2  tb0 14 0    LO1-S00-BH-J32 to LO1-N15
s 17 3  tb0 15 0    LO1-S00-BH-J34 to LO1-N16
# On board connections between switch chips on switch 1 in Frame LO1
s 14 7    s 13 4    LO1-S00-SC
s 14 6    s 12 4    LO1-S00-SC
s 14 5    s 11 4    LO1-S00-SC
s 14 4    s 10 4    LO1-S00-SC
s 15 7    s 13 5    LO1-S00-SC
s 15 6    s 12 5    LO1-S00-SC
s 15 5    s 11 5    LO1-S00-SC
s 15 4    s 10 5    LO1-S00-SC
s 16 7    s 13 6    LO1-S00-SC
s 16 6    s 12 6    LO1-S00-SC
s 16 5    s 11 6    LO1-S00-SC
s 16 4    s 10 6    LO1-S00-SC
s 17 7    s 13 7    LO1-S00-SC
s 17 6    s 12 7    LO1-S00-SC
s 17 5    s 11 7    LO1-S00-SC
```

Figure 46. Topology File Sample

The following is an example of how to interpret the lines in the topology file shown in Figure 47 on page 98.

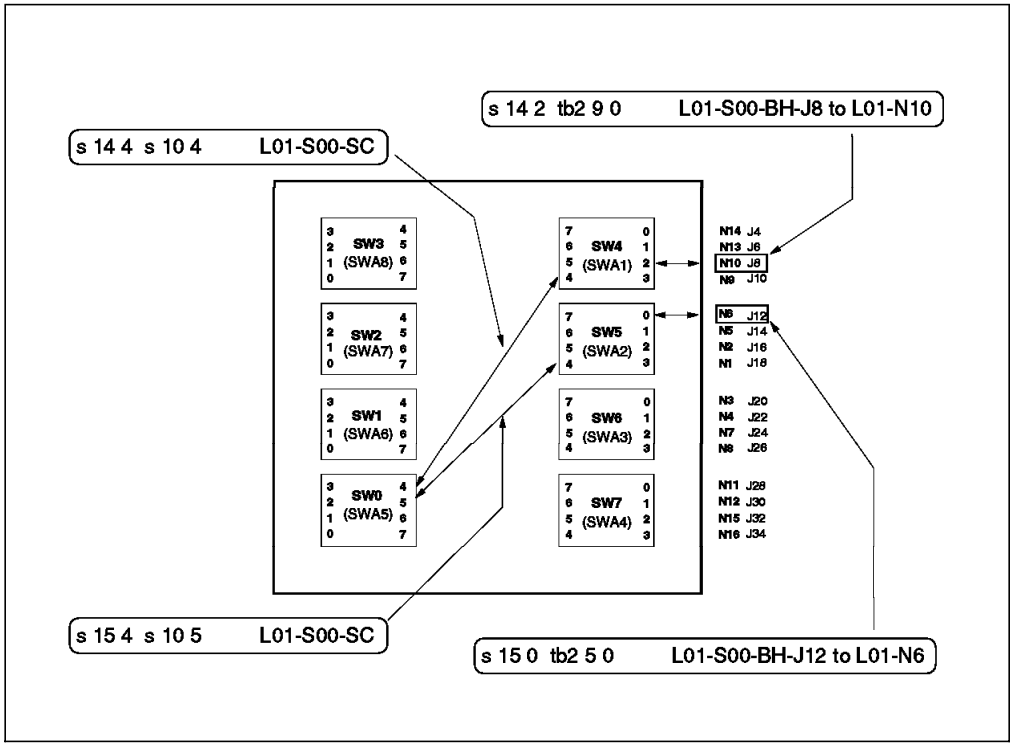


Figure 47. Topology File - Example

Comparing the HiPS and SP Switch boards, as shown, we see that the node-to-switch port connections for SP Switch are the same as for HiPS. The intra-switch connections are also the same. The only difference that would impact the topology file information is the jack (cabling) information. Table 6 summarizes those differences.

Table 6. Cabling - SP Switch versus HiPS

Chip	Node	HiPS Jacks	SP Switch Jacks	Chip	Node	HiPS Jacks	SP Switch Jacks
3	-	3	3	4	14	4	34
		5	4		13	6	33
		7	5		10	8	32
		9	6		9	10	31
2	-	11	27	5	6	12	10
		13	28		5	14	9
		15	29		2	16	8
		17	30		1	18	7
1	-	19	11	6	3	20	26
		21	12		4	22	25
		23	13		7	24	24
		25	14		8	26	23
0	-	27	19	7	11	28	18
		29	20		12	30	17
		31	21		15	32	16
		33	22		16	34	15

4.3.2 Topology File Nomenclature

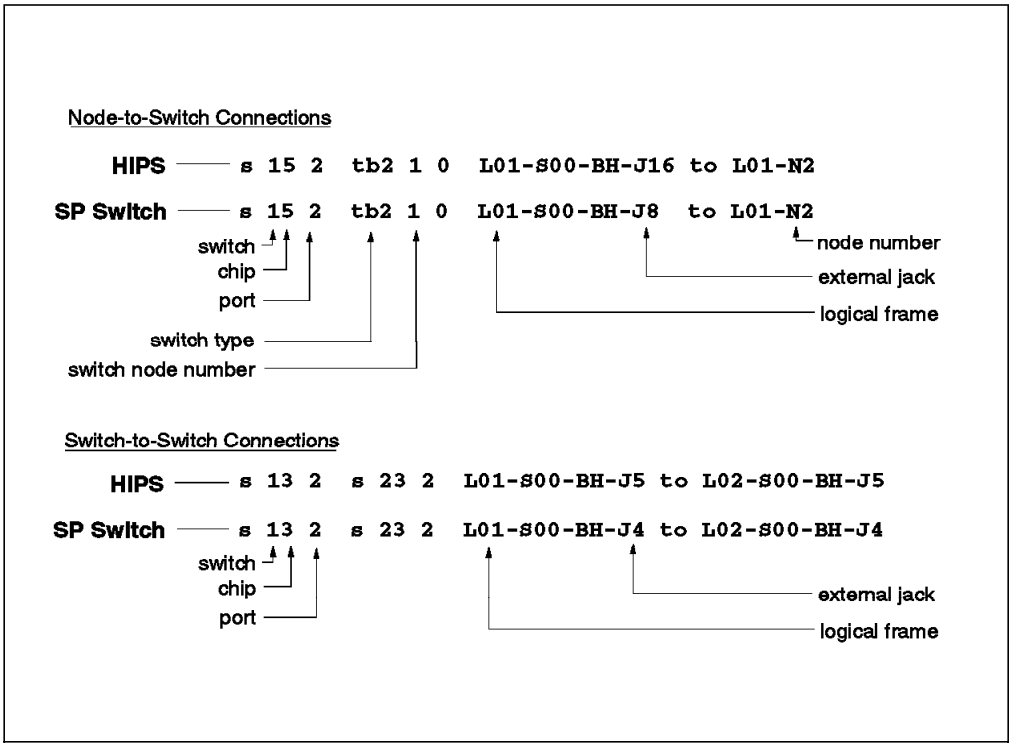


Figure 48. Topology File Nomenclature

Logical Notation

The logical notation has three fields that define one end of the cable and three fields that define the other end. An end of the cable is defined as <devices><logical number><port>, where:

- Devices are s(switch), tb0 or tb2 (adapters). The names tb0 and tb2 are generic for adapters and *do not* imply which type of adapter. It is actually more an indication of when the topology was put together for release (SP1 or SP2 time frame).
- The logical number for switch chips has as many as three aspects: (1) the last significant digit is the switch chip on the switch. This can be 0 through 7, because there are 8 switch chips; (2) the switch number; (3) if this is a switch in a switch frame, 1000 is added to the logical number to differentiate it from node frame switches. The switch frame switches are numbered 1 through the total number of switch frame switches. In other words, switch frame 1 houses switches 1 through 8, and switch frame 2 has switches 9 through 16, and so on. A number like 11 tells you switch chip 1 on switch 1. A number like 1023 tells you switch chip 3 on switch frame number 2.
- The logical numbering for the adapters begins with 0. Each number is tied to a specific node jack on a specific logical switch. If that node does not exist in the system, that logical adapter is simply ignored by the system. In the out.top log file, it will be noted as being replaced by a wrap plug.
- The port number for switch chips can be 0 to 7, because there are eight ports on a switch chip. Four ports leave the switch assembly and four ports go to the other chips within the switch assembly.
- The port number is always 0 for adapters.

4.3.3 HiPS Clock Subsystem

Understanding the switch clock subsystem will help you to get a better appreciation of how various failures in the clock tree could cause certain patterns of errors to arise in the log files.

- The HiPS clock subsystem

  The clock subsystem is divided in two parts: (1) The clock card and (2) The switch card. The clock card has an onboard oscillator and three inputs coming from the switch card data connectors that correspond to J3, J5, and J7 of the bulkhead. It redrives the selection of one of those four inputs. The redrive destination is the HiPS 3.0 card. There are also some redrives to J1 and J2 on the clock card. These were used by HiPS 2.0 switches (which, as you recall, have external clock cables), rather than embedded in the data cables.

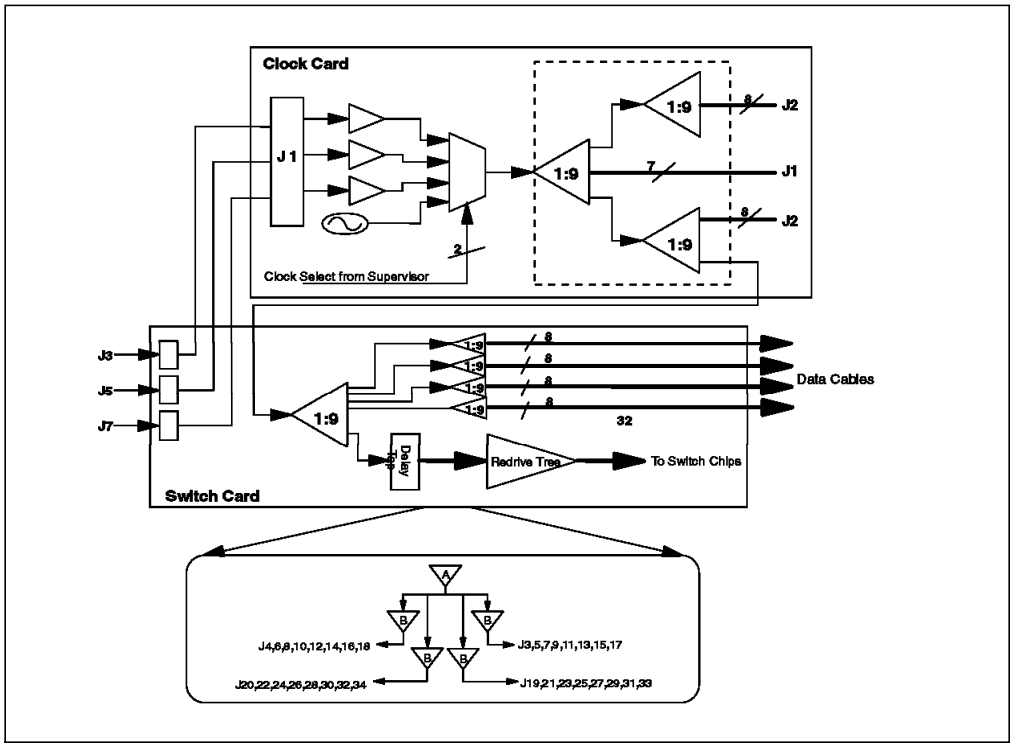


Figure 49. The HiPS Clock Subsystem

The clock is driven from the clock card selection logic to the switch card. From there, it is redriven to two major branches: one drives the data cables and the other drives the switch chips.

Each data cable has a clock put on pins 37 and 38. The cable twists these to the input pins 13 and 14 on the other side.

Of course, as mentioned earlier, only J3, J5, and J7 on the switch are used for the inputs, and each node gets a copy of the clock. The clocks that do not end up at nodes or at J3, J5, or J7 are terminated at the connector that receives them; there is a 100 ohm resistor across pins 13 and 14 of the data connector.

You can see how these are redriven by looking at the third part of the picture. From the clock card you have a branch at level A, which is divided into four branches in level B. Each branch is driven by a different redrive chip. Note that the odd jacks J3 to J17 are on one branch, while the odd

jacks J19 to J33 are on a different branch. If none of the jacks have a clock, then the problem most likely stems back to level A, or back to the clock card, or even back to the switch that is supposed to be driving that clock card.

Now, if none of the even jacks (J4 through J18) have clocks on them, which could be indicated by the diagnostics on all of the nodes connected to these jacks, you can be pretty sure that is the B level redrive chip that has broken. You can imagine the diagnostic power that you derive from knowing the clock tree.

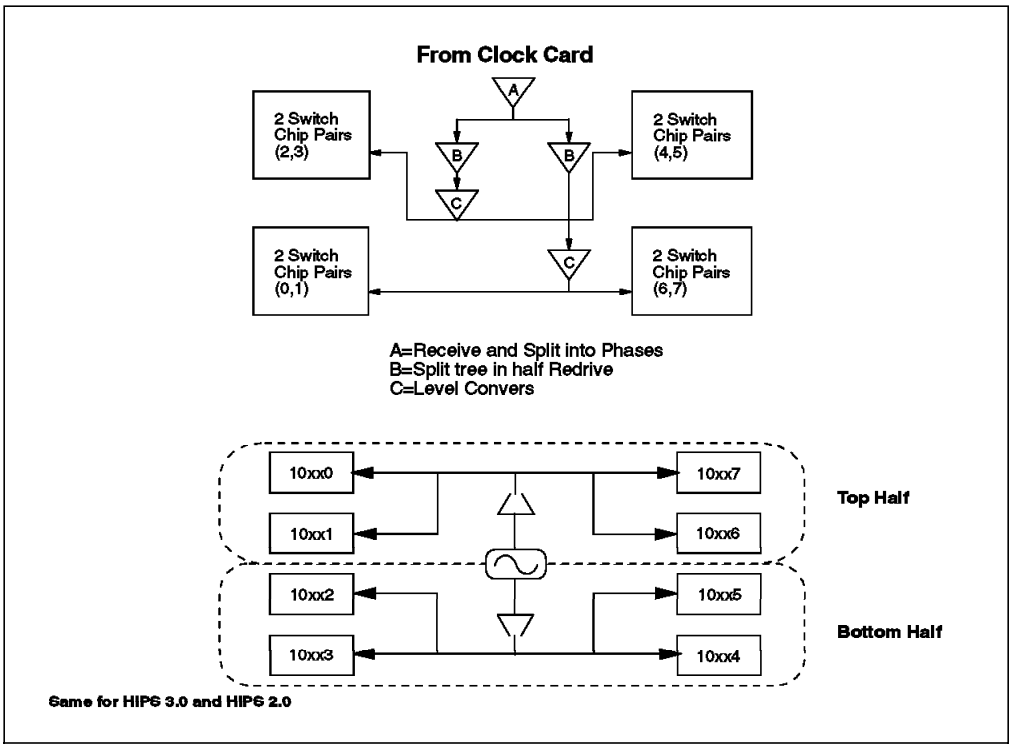


Figure 50. The HiPS Chip Clock Tree

First, the clock is put through a delay tap. The delay tap breaks the clock into eight phases, which are later used to tune the individual switch port. Each of the eight phases is driven to each of the switch chips. You can see how switch chip 0, 1, 6, and 7 are all redriven from one source and chips 2, 3, 4, and 5 are redriven from another. If you get failures trying to talk with all four chips on a single branch, and the others look okay, you can be pretty sure that the cause is the chip which redrives their clock.

Basically, when you get massive errors, look for patterns that match the clock redrive patterns, and you may be able to quickly narrow down your culprit.

4.3.4 SP Switch Clock Subsystem

The TBS board supports the use of multiple clock sources:

- Two internal oscillators (attached to two of the board′s
- One of two possible external (HiPS-like) clock inputs:
  - To be used for a center of the room clock source, or “dual switch” clock source, or both
  - Currently only one external connection is supported

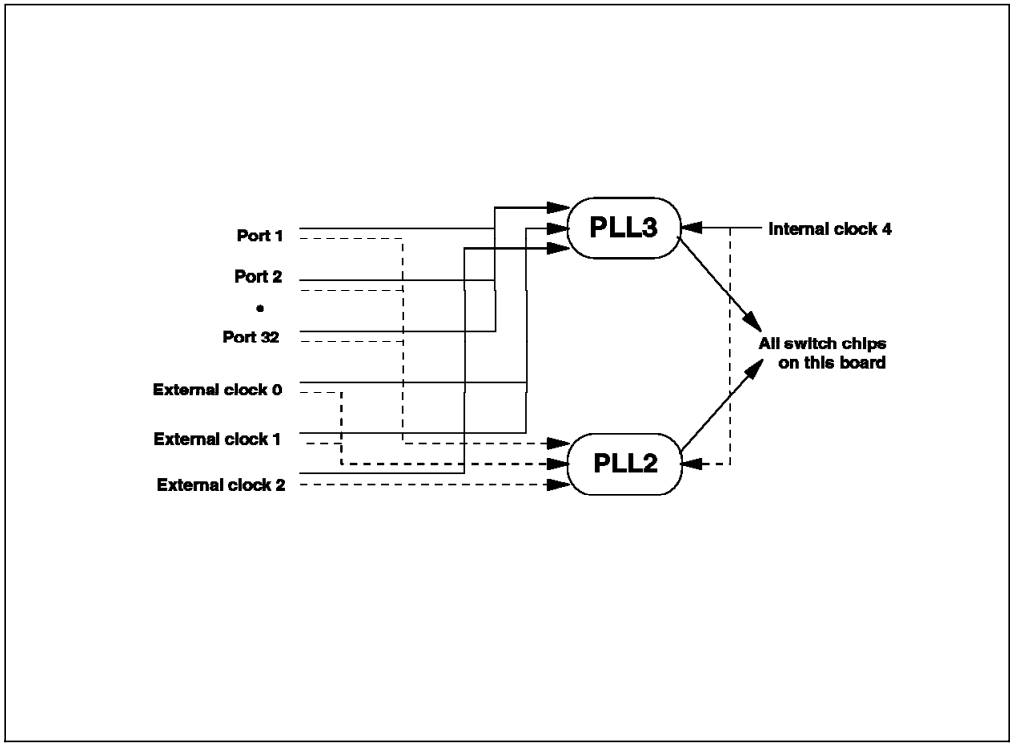


Figure 51. The High Performance Switch Board

Further, each phase-locked loop provides a clock redrive function, and any selected clock signal is passed through one of these redrives being distributed to all switch chips on board.

Each source must pass through one PLL, and for each PLL we have:

(2 internal) + (32 ports) + (2 external) = 36 choices per PLL

so there are (4 PLL)x36 = 144 total possible clock source paths. However, in the case of a port-sourced clock, the shortened systemwide clock-source path protects against excessive STI retime by restricting the (PLL, port)-pair usage to the “same chip.” (This minimizes the intra-board clock distribution path length for each non-master board.) Hence, the number of port choices is reduced to four for each PLL, and for a node switch board only non-node chips may be a source, yielding:

(2 PLL) x [(2 internal) + (4 ports) + (2 external)] = 16 total allowed choices.

Each one of the internal oscillators encompasses a clock redrive, and any chosen clock source must pass through one of these.

Currently only one external connection is supported.

4.3.5 Switch Clock Files

For the SP Switch, switch clocks in the system clock tree must be set in an “ascending level” fashion, with a reset of each switch along the way. Hence, a SP Switch Eclock topology file will contain a column indicating the level of the subject switch within the distribution tree.

The two types of reset which may be performed on a SP Switch are described in the following section. A POR is automatically performed when a switch is

powered. A *synchronous reset* is performed by the switch supervisor microcode each time the clock source (PLL or mux) is set.

- Power-on reset (POR)
  - Flush registers (zero all bits)
  - Self-test
  - Flush registers
  - Synchronous reset
- Synchronous test
  - Error enables off
  - Clear errors

Multiple clock tree alternatives will be available in an Eclock topology file, and so the successive alternatives will be presented one after the other.

4.4 Node Behaviors

The SP nodes can adopt different personalities when they are being part of the switch network.

4.4.1 Primary Node Behavior

The switch commands are issued on the Control Workstation. These commands tell the primary fault-service daemon to carry out the command.

The primary node considers the backup primary invalid if at least one of the following is true:

- The primary node has received an Error/Status packet from the backup′s switch port, and subsequent attempts to contact the backup fail.
- The primary node has received a bad Error/Status packet from a scan of the network, and subsequent attempts to contact the backup fail.

If the primary detects the backup has failed, it will automatically bring up a new one. The new backup will be chosen such that it is as “far” away from the primary as possible, in terms of the switch and physical node location (that is, on a different switch). The intent is to make the probability of simultaneous primary and backup failures as low as possible.

When a switch chip detects a condition the primary should know about, it sends both Error/Status packets to the primary, to virtually guarantee that the primary receives the Error/Status packet at least once. The two Error/Status packets take routes that are disjointed, with the exception of the last hop, which links to the primary node itself.

4.4.2 Primary Backup Node Behavior

The primary backup node has two duties. First, the primary backup node detects the loss of the primary node and initiates the takeover processing. Additionally, but usually the duty it is performing the most, the primary backup node functions as any other secondary node.

The backup daemon will start a thread that will queue a request token if it has not received a packet from the primary in M seconds, where M = 300 by default, and it is called in the primary′s scan interval. When the daemon receives the

token, it will verify if it is still the backup before it takes over for the primary. The backup daemon cancels this thread when it is no longer the backup.

4.4.3 Secondary Node Behavior

Every node, including the primary and the backup primary, executes the following steps in order to start up the switch:

1. Builds the "ideal" database based on the topology file
2. Modifies the "ideal" database based on "deltas" received from the primary node
3. Calls the route table generator to build the routes from this node to all other nodes
4. Tells the switch protocols (IP and User Space) which destinations are (or are not) available
5. Loads the routes down to the adapter

The deltas are messages indicating uninitialized/fenced nodes and/or uninitialized/fenced links received from the primary node.

For the most part, the secondary node repeats steps 2 through 5. The first step applies only on IPL or when the topology file is changed (such as with switch partitioning).

Secondary nodes send/receive service packets to/from the primary node only: The secondary node acknowledges the primary node for each database delta message it receives from the primary node.

4.4.4 Recovery from a Switch Failure

If a switch fabric or node failure occurs, the primary node must log the incident, and decide whether the active configuration must be adjusted to render the "ill" component(s) inactive. If the failing component is a switch link, the route table must be updated so the faulty link is no longer used. If a non-primary node is no longer reachable, the primary will fence that node.

If the primary node fails, the primary-backup must "take over." It must:

- Fence the primary
- Assume the primary personality
- Choose a new primary
- Run switch initialization

4.5 Switch Commands

Eannotator

- Annotates the connection labels onto the topology file.

Figure 47 on page 98 shows some of the connection labels in a topology file. Below is an example of a connection label:

```
L01-S00-BH-J18 to L01-N1
```

The system only uses these labels when reporting errors in the error log or the css logs. If these labels are incorrect, then it will be difficult to debug problems because the errors may be attributed to the wrong node.

Eannotator physically checks which connections it is going through (that is, the jack socket or the switch chip connection) to get to a node or switch chip and will update the topology file accordingly. It converts what is initially logical information to actual physical information. For example, L06 for logical frame 6 may be converted to E8 because this may be the 8th physical frame.

After migrating to the SP Switch, it is essential to run Eannotator after the hardware installation because the jack connections have changed completely from those which were used for the High Performance Switch. The system does not use this information during initialization, needing only the *logical* data to the left of this in the file which describes each link on the network.

Eannotator is run after the topology file has been selected as part of the installation process.

Etopology

- Stores or reads a switch topology file into or out of the SDR.

After the switch topology file has been annotated, it will be stored to the SDR by Etopology during the installation process. The switch topology file stored in the SDR can be overwritten by having an expected.top file in /etc/SP on the primary node.

Estart

- Starts the switch

If Estart can find the file expected.top file in /etc/SP on the primary node, it will use that to initialize the switch. Otherwise it will transfer the one stored in the SDR on the Control Workstation to that directory on the primary node. Estart_sw is run on the primary to initialize all the nodes in the switch. For the High Performance Switch this process will then clean up the topology file that was placed in its /etc/SP directory so that a new one will get transferred at the next Estart.

If the switch fails to initialize all nodes for any reason, always check to see if there is a /etc/SP/expected.top file on the primary node. If there is, check to see if it is the correct topology file, and that its contents are valid. For the High Performance Switch the only reason one of these files should exist is if an IBM engineer wishes to temporarily override the topology file in the SDR to assist in the diagnosis of a possible hardware failure on the switch. Once that work has been completed, the file should be removed and the SDR file should be used. Check to see if any of these files exist on any of the nodes by running:

```
# dsh -a ls -l /etc/SP/expected.top
```

If one of these files exists, then check it out because if the primary is changed to that node, it may affect the initialization of the switch.

For the SP Switch, this procedure is different. A check is made to see whether there has been a change in the switch topology by referencing the SDR, and if this is not the case it uses the existing topology that has been previously

distributed. The cleanup of the /etc/SP/expected.top is not carried out for this switch. If there has been a change, then the new topology file is copied into /etc/SP on the primary. This topology file is further distributed on to the nodes (through the boot/install servers, if there are any).

Eclock

- Controls the clock source for each switch board within a RS/6000 SP.

The clock source topology file (located in /etc/SP on Control Workstation) is selected during installation of the system. Eclock uploads the mux values to each switch board, which tells it where it will get its clock source, and then does a reset on the board to resynchronize the switch chips. This is disruptive to normal operations on the switch and care should be taken when this command is run so that data is not corrupted or lost.

Efence

- Removes a SP node from the current active switch network.

For the High Performance Switch after a request to fence a node is issued, the switch adapter is flagged that the request has been made to come off the switch network. The Worm on the primary node is notified by a service packet during service phase (see Eduration below), and it creates a new routing table based on the fact that the link to the node is not included. The Worm on the primary then distributes these to the nodes. In this way the link to the node is disabled. The nodes are not able to generate a switch fault by sending a packet to the fenced node, and so the switch fabric does not get disrupted.

With the SP Switch, the routes are not distributed over the network; only the changes to the device database are. This database contains a listing of the active nodes for that partition (that is, it does not include nodes that are fenced or off the network). If a node is fenced with the autojoin option, then it is included in the list of active nodes because it is ready to rejoin the network at the next Estart. Any disabled links will be also flagged in this database. The primary Worm will then generate a new routing table to download onto its SRAM on the TB3 adapter. It will communicate the changes in the database as a delta (that is, it will not send the entire database) to the other nodes. The Worm daemons on the nodes are able to compute the routes locally for loading onto the SRAM on the switch adapters.

Eunfence

- Adds a SP node to the current active switch network that was previously removed with the Efence command.

Efence and Eunfence allow you to plan maintenance on a node and smoothly take it on- and off-line with minimal effect on the switch network. By default, every two minutes the switch network goes into service phase and looks for a node to fence or unfence (see Eduration).

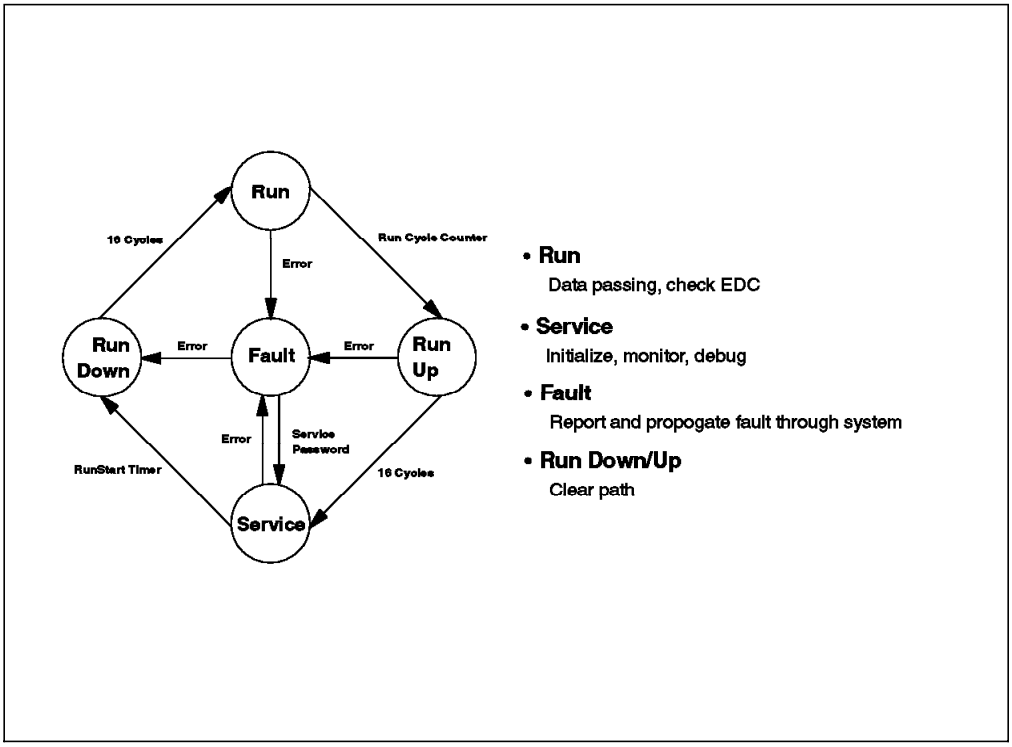


Figure 52. Diagram of the Different Phases on the High Performance Switch

Read the section on switch logs and also the following information, which gives some indication of how this problem has occurred.

There are multiple ports called out by switch chips in the /var/adm/SPlogs/css/flt. This rarely happens on real faults. When the chip can narrow down to a single port, there is usually only one port that breaks. The exception would be when multiple ports on a chip are connected to the same switch, which has been powered off. This indicator is particularly interesting when you start seeing random ports being reported over time (for example, ports 0, 1, and 7 reported at one time and ports 0, 1, and 6 reported at another time). Another strong indication of this problem is that multiple onboard switch ports (ports 4, 5, 6 and 7) are reported. Recall that these connect to different chips on board.

The adapters may report MSTAT=12345678 or 000000080.

A message like: `run_phase_transition: message not sent in time,` may be generated. The next fault seen is likely to be caused by this timeout.

If there is a problem with switch faulting and there are symptoms like those described above, then contact the local IBM Support Center and mention APARs IX53234 and IX54543.

Additional Information

Important

This applies only to the HiPS. The new SP switch does not have a service phase.

4.5.1 The rc.switch Script

The rc.switch is run from the /etc/inittab during boot.

Whenever there is a problem initializing any of the nodes onto the switch, check that the Worm daemon is running on the relevant nodes:

```
# ps -ef | grep Worm
```

If it is not running, then start it by running rc.switch script and check to ensure the daemon stays up. If the Worm daemon is not running on the primary, then none of the nodes will be initialized.

Always check the /var/adm/SPlogs/css/rc.switch.log to see any abnormal behavior.

The content of various switch log files will be explained later in this section.

The /usr/lpp/ssp/css/ifconfig command configures or displays the network IP interface parameters for the switch.

Always use this command in conjunction with the switch (css0 device) as modifications had to made over standard TCP/IP implementation of AIX for this network. Run the following command on the nodes to check that the *css0* interface is up and all the nodes have the same attributes.

```
# dsh -a /usr/lpp/ssp/css/ifconfig css0
```

Ensure that there no inconsistencies. For example, when a node does not have ARP enabled when others do, it may not be possible to even *ping* that node. Run the following command to resolve this on that node:

```
# /usr/lpp/ssp/css/ifconfig css0 <switch IP address> arp
```

A detailed flow chart of this script can be found on Appendix A, “RS/6000 SP Script Files” on page 229 under A.4, “The rc.switch Script” on page 262.

4.5.2 Switch Initialization

/usr/lpp/ssp/css/Estart_sw is called by /usr/lpp/ssp/bin/Estart script, and is run only on the primary node (oncoming for SP Switch). It is useful to know that the timeout variable *LIMIT* for the *Estart* command is located in this file. This may have to be adjusted for systems that take longer than the default (180 sec).

Related Problem That Has Been Experienced:

The daemon must have enough time to go around and touch all of the switch resources in the system. Due to heavy load on the primary node, there is a possibility that the daemon will not finish the job within a timeout window. If the timeout is reached, the switch network may experience switch faulting. The problem has been experienced on systems that have primary nodes with heavy Micro Channel loading. Also, the larger the system, the more time it takes to get around the network. In this case, the probability increases that the timeout will be reached.

Later in this section we will discuss how to diagnose this kind of problem by analyzing the switch log files.

Notice that in the SP Switch, the *Estart_sw* calculates the timeout based on the number of switches:

```
LIMIT = 180 + ((NUM_SWITCH - 2) * 60)   where NUM_SWITCH > 2
```

The following steps are referred to as *Switch Initialization*:

1. An *Estart* is issued for a system partition from the Control Workstation, through either SMIT or the command line, and this is relayed to the primary node of the partition as an *Estart_sw*.
2. The primary node consults the SDR for the current topology file and the current fenced nodes.
3. The primary node distributes the topology file as appropriate:
   - The primary node distributes the topology file to the bootserver nodes of its partition.
   - The primary node directs each bootserver to distribute the topology file to its client nodes.
   - Each bootserver distributes the topology file, tests for success, and returns resulting status to the primary.
   - The primary node logs and evaluates failure data.
4. The primary runs the Worm code to verify the partition's fabric pool and nodes:
   - Each chip is sent a *Read Status* service package to check cabling.
   - Each chip is sent an *Initialization* service package to set the chip ID and specify routes to the primary.
   - Each non-primary unfenced node is sent an *Initialization* service package to set its personality and specify the topology file for this partition.
   - Each non-primary unfenced node is sent a *Read Status* node service package to verify the switch address, personality, and readiness.
   - The primary node updates its device database in accordance with the findings.
5. The primary node sets the Time-of-Day (TOD) across the fabric pool:
   - Traffic is quiesced, if necessary.
   - The TOD is propagated.
   - Traffic is resumed, if necessary.
6. The primary node downloads its route table to its TB3 adapter
7. The primary node distributes device database deltas to the nodes over the switch through the *Device Database* service package
8. On each non-primary, unfenced node:
   - The local device database is updated with the deltas.
   - Routes are computed.
   - The routes are loaded to TB3 SRAM.
9. On each node, the communications protocols and Load Leveler code, as appropriate, are notified that the switch is available.

10. The primary node updates the SDR *switch_responds* class for its partition. For each node in the partition, the (*autojoin,isolated*)-pair is set to one of the following:

    (0,0) Initial

    (0,1) Fenced

    (1,0) On

    (1,1) Suspended

4.6 Reviewing Switch Processes

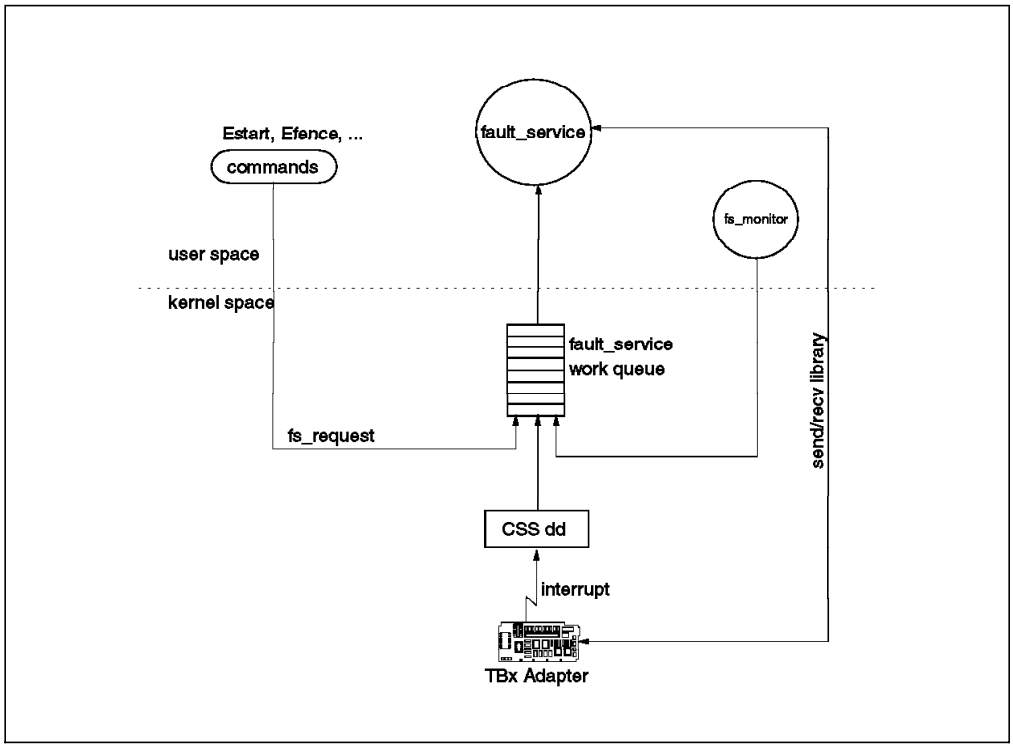


Figure 53. The High Performance Switch Board

fault_service_Worm_RTG

The Worm daemon is central to the functioning of the switch. It can be viewed as having two main personalities:

- Primary Worm
- Secondary Worm

There is a third personality with the SP Switch:

- Primary backup Worm

The primary Worm has a key role to play in the coordination of the switch network. It is responsible for generating and updating the route tables through the Route Table Generator(RTG), and for handling and recovering from switch faults generated on the network. Switch faults are generated for a number of reasons. For example, corrupted data packets, hardware faults, and running

Estart can all create a switch fault. Either the Worm on the nodes or the switch chips themselves can report a fault to the Worm on the primary node.

The Worm process verifies the switch connections beginning at a node designated as the primary node. By default, the primary node is node 1 in frame 1. You can override the default and designate another node as the primary node. With the High Performance Switch, this must be done manually if the primary is not operational, and the Estart command should then be run.

fs_monitor

This daemon monitors the adapter for interrupts that have not been serviced.

To check that both daemons are running on a node run:

```
# ps -ef | grep css
```

Note: Add "_SP" to each name to find out the SP Switch daemons.

4.6.1 Switch Responds

The spmon GUI for switch_responds has changed for the SP Switch. Figure 54 shows the differences between them:

	High Performance Switch	SP Switch
initial state	red	yellow
on the switch	green	green
fenced (no autojoin)	yellow	red
fenced with autojoin	yellow	yellow
failure		red

Figure 54. Switch Responds

For the SP Switch, an extra attribute has been added to the SDR object class *switch_responds* called *adapter_config_status* which gives the results of the switch adapter's configuration method. This field is left blank for the High Performance Switch adapters only if you do not have PTF 11 installed. Here are the possible values for this attribute for the SP Switch at the present time:

- not_configured (initial value)
- css_ready (method was successful)
- odm_fail
- chkslot_fail
- busresolv_fail
- getslot_fail
- xilinit_system_fail
- xilinit_load_fail
- dd_load_fail
- genmajor_fail
- genminor_fail
- make_special_fail
- build_dds_fail
- dd_config_fail
- diag_fail
- fs_load_fail
- fs_init_fail

If a node does not come up on the SP Switch, always check this attribute using the following command:

```
# SDRGetObjects switch_responds adapter_config_status
```

This may give a good indication of where the problems lie. To assist with the understanding of these attributes, here is a list that summarizes the configuration of the TB3 switch adapter for the SP Switch (cfgtb3):

1. Checks the ODM database for the *css* device (odm_fail).
2. Loads the xilinx microcode for the adapter (xilinx_system_fail and xilinx_load_fail).
   - There are distinct xilinx loads for the 2 types of adapter
     - tb2_xilinx_file
     - tb3_xilinx_file
3. Loads the device driver (dd_load_fail).
   - There are distinct drivers for the depending on the type of adapter
     - cssdd2
     - cssdd3
4. If the special device file /dev/css0 does not exist, then it creates it (genmajor_fail, genminor_fail and make_special_fail).
5. Calls the device driver to configure the device (build_dds_fail and dd_config_fail).
6. Marks the device css0 as available.
7. Executes POST diagnostics with the different routines depending on the type of adapter (diag_fail).
8. Loads and initializes the CSS kernel extension which provides support for fault service and DMA (fs_load_fail and fs_init_fail).
9. Sets *switch_responds* attribute *adapter_config_status* to *css_ready* in the SDR (for TB3 only).

The three attribute values not explained so far are:

- chkslot_fail
- busresolv_fail
- getslot_fail

After the check is made to test whether the ODM attributes are present for the adapter, some tests are carried out on the bus to detect which slot the adapter is in.

From the value in this object class, it is possible to find out the area in which the failure occurred. This is a major diagnostic benefit for the TB3 adapter.

For completeness, here is the *unconfigure method* (ucfgtb3):

1. Performs the ODM check for the TB3
2. Disables the TB3 adapter
3. Unloads the CSS kernel extension
4. Kills the CSS daemons
5. Unloads the TB3 device driver

6. Marks the device *css0* as *defined*

There is an undocumented command called `css_restart_node` that runs first the unconfigure method and then the configure method. This command should only be run at the request of an IBM engineer as the results can be unpredictable in certain circumstances.

4.7 Switch Log Files

There are numerous log files associated with the switch, some of which can provide essential data for resolving switch problems. It is impossible in this redbook to provide enough information about how to read and interpret these files. Instead, a brief example is given of what type of data is contained in these files, along with an example of some typical output. It is not expected that the reader will be able to resolve all the problems by using these logs, but that the first level of analysis can be undertaken in most cases prior to contacting the local IBM Support Center. A utility is provided called `css.snap` which will create a compressed tar image of all the switch logs in the following format:

```
/var/adm/SPlogs/css/<hostname>.<date>.css.snap.Z
```

Run the command:

```
# dsh -a /usr/lpp/ssp/css/css.snap
```

Then transfer all the compressed tar images to a single system such as the Control Workstation to provide the data to the local IBM Support Center.

/var/adm/SPlogs is a specific directory for SP logs. In this directory you will find the following files:

On the Control Workstation:

- /var/adm/SPlogs/css/Eclock.log
- /var/adm/SPlogs/css/Ecommands.log
- /var/adm/SPlogs/sdr/SDR_config.log
- /var/adm/SPlogs/sdr/sdrlog.<cws-ip-address>.<PID>

On the Primary Node:

- /var/adm/SPlogs/css/rc.switch.log
- /var/adm/SPlogs/css/out.top
- /var/adm/SPlogs/css/fs_daemon_print.file
- /var/adm/SPlogs/css/flt
- /var/adm/SPlogs/css/dtbx.trace
- /var/adm/SPlogs/css/dtbx_trace.failed
- /var/adm/SPlogs/css/daemon.stderr
- /var/adm/SPlogs/css/daemon.stdout
- /var/adm/SPlogs/css/cable_miswire
- /var/adm/SPlogs/css/router.log
- /spdata/sys1/logtables/css.tab

On a Node:

- /var/adm/SPlogs/css/rc.switch.log
- /var/adm/SPlogs/css/dtbx.trace
- /var/adm/SPlogs/css/flt
- /var/adm/SPlogs/css/fs_daemon_print.file
- /var/adm/SPlogs/css/daemon.stderr

- /var/adm/SPlogs/css/daemon.stdout
- /spdata/sys1/logtables/css.tab

rc.switch.log

The log file contains specific information about the node that relates to the switch. If you do not know what kind of switch type is installed at the SP System, look at this log file. The information is retrieved from the ODM. The rc.switch script configures the Switch adapter *css0* by issuing an `ifconfig` command. Any error relating to the `ifconfig` command will also appear in this log file.

This file is updated/created when rc.switch has been executed. Usually it is run from /etc/inittab at boot time. The rc.switch script starts the fault_service_Worm daemon as well as the fs_monitor daemon.

out.top

This file will show problems that arise during switch initialization (*Estart*). A copy of the out.top file of a previous run will be called out.top.old. Both files will only be found on the primary node. It is like a copy of the topology file.

Example:

```
Exx-Nxx -17 R: link to device is replaced by wrap plug or cable
```

This error usually indicates that this is an empty slot in the frame caused either by having an installed wide node or a partially filled frame. The *-R* indicates that the error was on the right hand side of the link, which in this example is the node or switch adapter side rather than the switch chip port side.

When `Estart` finds problems with the SP Switch, they are reported next to the switch connection that has the problem. Most of the error messages are documented in the Messages and Diagnosis Guide.

If Estart fails to initialize any of the nodes, go to this file and see whether there are any error messages which could indicate the cause of failure. If there is an unusual pattern of errors where it reports wrap plugs where they should not be, it could be that there are problems with the configuration. Go back and verify that the data is all correct and consistent.

fs_daemon_print.file

This log file is a detailed account of the fault_service_Worm_RTG daemon.

If the Worm keeps dying, look into this file to attempt to find an indication of why this is happening.

flt

This log is a cumulative record of switch faults, including those generated by an `Estart`. Normal usage of the switch may cause occasional faults. This file is not only an error log.

An indication of an adapter problem may be multiple occurrences of switch faults for the same node, which would show up in this file. An indication of a switch chip problem will also be shown with multiple switch faults in this file.

The format of this file is considerably different between the SP Switch and the High Performance Switch. This in part reflects the difference between the two switches in the way they detect and handle switch faults.

dtbx.trace

This provides traces from switch diagnostics.

dtbx_failed.trace

This is created if any of the switch diagnostics fail. It is basically the same as the dtbx.trace file, with the addition of error messages. If the diagnostics run clean, this file will *not* be created.

daemon.stderr

General error messages. For example, if some nodes could not get initialized when the Estart was issued, you may see the following:

```
Processing Estart. The following node(s) could not be initialized:
  sp21n03
```

In this case the node in question (sp21n03) was not initialized because its Worm daemon was not running.

daemon.stdout

This log is a detailed account of the switch initialization process. Most switch problems do not require analysis. However, in some circumstances it may prove useful to diagnose the problem.

There are also many normal, informational switch messages that you will see in the error report. For instance, when you issue an Estart, you will see a switch fault in the error report.

Expect one of each of these messages for each Estart command:

```
ERROR_ID  TIMESTAMP T CL Res Name   ERROR_Description
34FFBE83 0502140496 T H  Worm       HPS Fault - detected by switch chip
C3189234 0502135796 T H  Worm       HPS Fault - not isolated
```

In addition to these log files, there is always a possibility that errors will be recorded in the error report (*errpt*). It is important to note the time of the failure in order to correlate messages in the error report *errpt* to the error.

For many problems, looking in the log files does not provide sufficient information to solve the problem.

In the directory /usr/lpp/ssp/css, there are a few helpful tools, such as:

css_dump

This will format trace entries relating to the cssdd. To run the command, issue css_dump > /tmp/css_dump.out &. You will find the most recent entries at the bottom of the output file. This information is helpful in conditions where the *css* driver code hangs for unknown reasons. This command should be run on the primary node and on any of the failing nodes.

fs_dump

This will format fault_service kernel extension traces. This command should be run on the primary node and any of the failing nodes. To run the command issue `fs_dump > /tmp/fs_dump.out &`.

4.7.1 The out.top Log File

The following is an example of out.top file output:

```
s 14 2  tb0 9 0         E01-S17-BH-J32 to Exx-Nxx
s 14 2  tb0 9 0         E01-S17-BH-J32 to Exx-Nxx -4 R: device has
been removed from network - faulty (wrap plug is installed)
```

The out.top log file is broken into a logical and physical notation. There are also comments in the file that are designed to help you understand the basic connections that are occurring. These comments are preceded by the pound sign (#). The comments at the top of the file should help you to remember the format. Other comments describe a group of connections (such as nodes connected to board 1). These comments precede the connections which they describe.

After the logical and physical nomenclature is given on a line, the fault information is given with an error number, an indication of which side of the connection found the error, and a description of the error: `-4 R: device has been removed from network - faulty (wrap plug is installed)` means there is a wrap plug where an adapter was expected, usually because the previous was a wide node.

out.top messages

1. -1 Indicates that a switch resource is uninitialized. It says that the initialization code never even got far enough in the network to attempt to initialize the resource. Something was broken downstream from it that prevented the code from getting to it. Traditionally this has only been done for nodes. However, this should change in the future. This is quite often an indicator of some sort of clock problem.
2. -1, -3, -4 are faulty link indicators. When these are reported on single ports, they can indicate problems with a cable or a port that will lead to FRU calls. It can also indicate a powered-off node or switch. Patterns of these can also indicate other problems with clocks and power.
3. -5 is rare message that indicates you should look for the cable_miswire file on the primary node.
4. -6 indicates some kind of problem with an adapter. This would be an unusual occurrence.
5. -7 indicates that for some reason the fault service daemon on the node could not respond back to the primary within the time limits. This quite often means that the daemon was killed or never started. The rc.switch.log or fs_daemon_print.file log files on the node can give you indications as to why this happened. There is a rarer condition in which the node is so busy that the daemon does not have time to respond. (This was more prevalent in SP1 days.)

6. -8 is the most common indicator of a miswire. You should look for a cable_miswire file on the primary node.
7. -9 could mean that the switch element (chip or adapter) was failing.
8. -10 through -15 are indicators that something failed during testing or a switch chip′s RAM. The switch board on which it resides is the FRU.
9. -16 usually indicates some sort of clock setup problem. This is most often reported on the nodes. For some reason the switch chips are usually reported with a -2, -3, or -4 when they have clock problems. If you see it on a node, what usually has happened is that the node was powered-on before the switch had its clock set up to the correct clock. In the case of a single switch system, the node was powered-on before the switch. There was also an old bug where the power-on diagnostics selected the wrong clock on which to run: a gore clock instead of a data cable clock when attached to an HiPS 3.0, or a data cable clock instead of a gore clock when attached to an HiPS 2.0.
10. -17 is more of a warning than an error. Make sure that it makes sense that there is a wrap plug on that port. It makes sense when the node is not populated or when the cable has been removed for some reason.
11. -18 is similar to -17, but occurs much more rarely.
12. -19 is very similar to -7 and was introduced later in the program when we began distributing routes across the switch. The fault service daemon on the receiving node must have time to service the route table information.
13. -20 is similar to -8, but is rarely seen.
14. -21 and -22 are very strange and have never been seen. Call for help!
15. -23 will be seen more often when more nodes are fenced in a system.
16. -24 should never occur. If you see it, call for help.

This file is very useful, because this is a dump of the topology file with some comments about errors found when the switch was initialized. Looking at this file, you can find at a glance some common wiring and node down problems.

4.7.2 Patterns in out.top File

1. Uninitialized = -1 -> look for pattern
2. Link errors = -2, -3, -4 -> look for pattern
3. Miswires = -5, -8, -20 -> check cable_miswire
4. Node time-outs = -7, -19 -> look for pattern
5. Unusual which imply FRU immediately = -6, -9 through -15 -> replace FRU
6. Clock = -16 -> look for pattern
7. Warnings = -17, -18, -23 -> verify the statement it makes is expected
8. Call for help = -21, -22, -24 -> call

Categories that require pattern recognition: 1, 2, 4, and 6.

Patterns:

1. An internal port (ports 5, 6, 7, 8) is reporting problem

For the most part, if an internal port reports an error, the FRU is the card that contains that port. You may discern patterns that indicate clock problems, but you will quickly deduce that the FRU is the same because the clock problem is a broken driver on the switch card.

2. All board patterns

   There should be very few errors that occur on the whole board, because once you get enough errors, you cannot get to the other parts of the board to see if they are in error.

3. Half board pattern

   When you see a half board with a problem, check to see if it matches the clock tree. The majority of these patterns of error are clock-related.

4. All the nodes are reporting a problem

   If all of the nodes are reporting an error, it is usually a -1, -7, -16, or -19. These quite often point to clock problems, or power sequence problems, which result from nodes being on the incorrect clock.

   A -1 on all of the nodes on a board will occur on boards that are not on the same board with the primary node: the primary node throws this off, because if you have gotten as far as generating an out.top file, the primary node is operational and reachable.

   With a -1 on all the nodes, you will probably see -2, -3, or -4 on the ports that connect to other switches. What this indicates is that the failing switch board is not on the same clock as the primary node′s switch board. This can happen because:

   a. The clock tree was not set up properly by issuing an *Eclock* with the correct Eclock topology file.

   b. The clock to this board is broken in this switch assembly, in the cable, or in the switch that sources this board′s clock.

   c. If the primary node′s board is the only one that is tunable, it may be set to the incorrect clock.

   A -16 can indicate a poor power-on sequence. For example, if the nodes in a frame are powered-on before the switch, all the adapters will be set to their internal clock (on card oscillator). You can discover this by looking in the dtbx.trace file. Another example is that the switch was powered on before the nodes, but the clock was not properly set. When the proper clock is selected, the clock is momentarily interrupted, which causes the adapter to have problems.

   -7 and -19 both point to daemon time-out problems. The -7 occurs when the primary node has initialized a node′s adapter and is waiting for that node′s daemon to respond to communication that asks what node it thinks it is. The -19 occurs after the whole switch has been tuned and the route table is being distributed to the nodes. In this case, the daemon on the receiving node is not responding. These -7 and -19 errors can occur more easily in large systems than in small systems, because large networks will naturally eat more into the time-out period during normal operation than a small network will.

5. All of the switch-to-switch connections are reporting a problem

   This is quite often a clock or power problem. It can be on the board, the clock card, the clock source cable, the power card, or the power cable.

6. All the ports on a chip are reporting a problem

   If all of the ports on a chip are reporting a problem, and they are not all connected to the same switch board, it is most likely a problem with the chip. If they are connected to the same switch board, you will probably discover that all of the ports connected to that other switch have problems.

7. A single port is reporting a problem

   When a single port finds a problem, the maintenance manual is usually quite adequate in making the call.

   The error to watch out for in a single port pattern is the one where the primary node cannot get into the switch. This can point to a power sequence or clock problem on the primary node. Check the diagnostics file at /var/adm/SPlogs/css/dtbx.trace. You may find that it is on an incorrect clock. Of course, it could still be the adapter, the cable, or the switch port to which the primary node's adapter is connected.

4.7.3 The flt Log File

Important

This section and the following sections on the flt file apply only to the High Performance Switch.

Faults are not as important or critical for the SP Switch as they are for the HiPS switch, so the flt file has changed with the new switch, and it is no longer an important diagnostic tool.

It is very important to note that the flt file is not a diagnosis of the problem. It tells you *which device saw the problem*. The problem could be with that device, or it could be with the device to which it is connected.

What you are looking for in the flt file is a pattern of errors over time. One error in the flt file does not imply a problem. In fact, there is a finite probability that a fault will occur. What you are looking for is a device that continually faults, say every day, or whenever a certain application runs, or in some other consistent fashion.

When looking for patterns, you should realize that the flt file records all information that the fault daemon on the primary node collects from the switch. This includes nodes being powered off for service, or whenever someone runs Estart. You need to recognize such events so that you discard them from your pattern recognition.

- Estart is likely to have occurred when you see a VDC_FLT=0000 or 0080, and there is a report like this: No fault isolated, VSP VDC_FLT=0000.
- You must also be able to note when it is likely that a node was powered-off or a cable was pulled. If you see a fault reported when you know a node was powered-off, you should ignore it. If you see a fault that indicates a node might be causing the fault, you should check out the time stamp and see if the node was powered-off around that time.
- If you know that the system was in a state of turmoil during a certain period of time, you should suspect that faults during that time are to be expected. During such periods of turmoil, it is often the practice of system

administrators to power things off and on. Such events will cause switch faults.

- If many devices report errors at the same time, it is very possible that there is something strange going on. You should delve back into your knowledge of how the clocks are distributed and how the boards are connected before you determine if this is a real problem or not. You could be seeing power glitches, clock glitches or some sort of non-hardware related problem. For example, there is/was a problem with heavy Micro Channel loading on the primary node. In systems that are AIX 4.1 and beyond which have node isolation (switch fence/unfence) capabilities, the switch goes into service phase every 2 minutes to see if it needs to fence or unfence a node. If the daemon cannot get processor time on the CPU, usually because of heavy Micro Channel use, it will leave the switch in the service phase for too long, and a time-out will occur. This will cause massive faulting in the switch.
- If many ports on a switch chip report an error, there is probably something strange occurring with the clocks, the power, or again some sort of other bug like the one for fence/unfence.
- If an on board port (port 4 through 7) is called, the FRU is the switch assembly that houses the chip that is reporting the fault.
- If an adapter reports a fault and it has an MSTAT of 12345678, this is a bogus error. The software knows that it has done something that might cause an error to be reported; therefore, when it sees it, it flags that error as bogus by setting the MSTAT to 12345678.
- Link errors detected by adapters will have an MSTAT of 000000c0. This is because bit 6 of the MSTAT is the bit that indicates a link error has been detected.
- If an adapter reports a fault and it has an MSTAT of 00000080, you know that the error was found on the node side of the adapter. If this is accompanied by a VDC_FLT=1000, you should check the error log to see if a CRC error occurred. Basically, VDC_FLT=1000 and MSTAT=00000080 indicates that the software/microcode has found an error that it deems serious enough to cause the switch to fault. Quite often this is because of a CRC error, but there are other possibilities.
- If VDC_FLT is not 1000 and you get an MSTAT of 00000080, it is quite possible that the adapter or the IO planar have a problem; you should run diagnostics.
- If an adapter reports a valid link fault, you should run diagnostics.

4.7.4 Reading the flt File

Figure 55 on page 121 shows the flt file, which is discussed in the following sections.

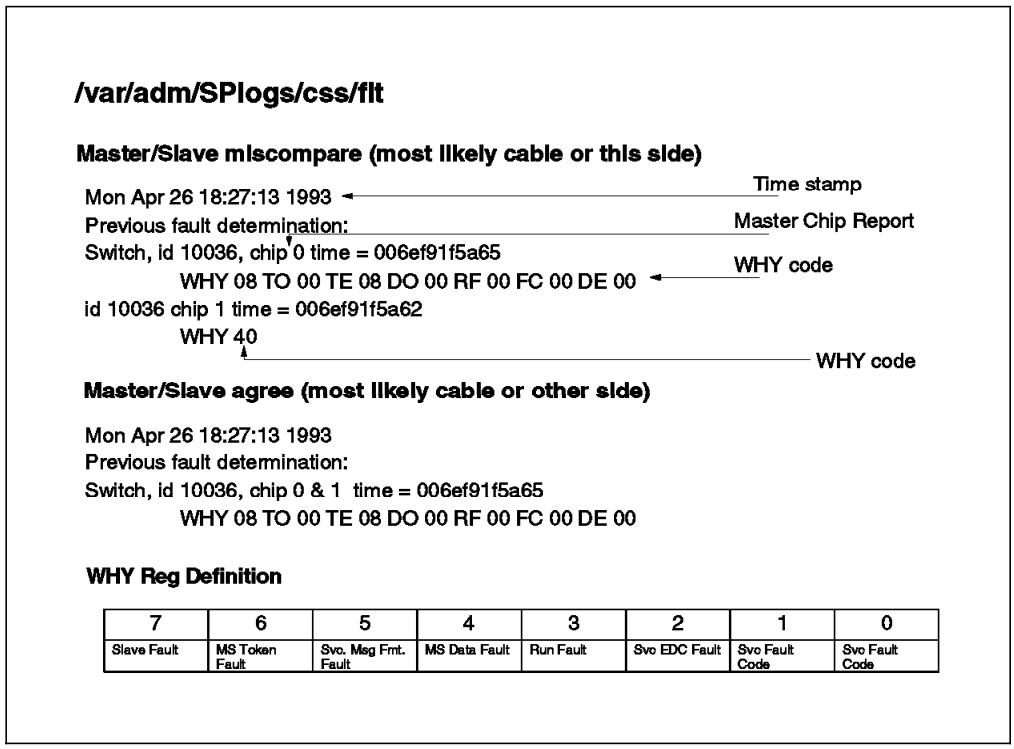


Figure 55. The flt File

The flt file has some nomenclature that differentiates device types, but it also lists the device number in a field that is generically labeled “id.” With this in mind, it was decided to differentiate switch IDs from nodes IDs by adding 100,000 to them. In this way, you cannot confuse a node with a switch, unless you have more that 10,000 nodes in your system. Why 10,000 nodes? The last digit of the switch ID tells you the chip on that switch. So, you are actually adding 10,000 to the switch board number. This leaves 10,000 node IDs before you hit the first switch ID at 100,000.

Note

- Therefore if the ID is < 100,000 you know that it is an adapter.
- If the ID > 100,000 you know it is a switch.

You can translate the flt notation to out.top notation so that you can look in the out.top file to determine the physical location of the connection.

The important information in the flt file is:

- The device ID, which tells you which device reported seeing the fault.
- For switch chips IDs, the WHY registers information. This is really important to determine master/slave miscompares and which port is reporting the problem. The actual error type in the WHY is of lesser importance. If the chip can narrow down the information to a port, it will report WHY xx TO xx TE xx DO xx RF xx FC xx DE xx, where TO, TE, DO, RF, FC, and DE are registers for each different error category underneath that particular WHY code. The register value, xx, indicates which port saw that error. Each bit stands for a port. Port 0 is the least significant bit, and port 7 is the most significant bit. If a particular error, such as TO, was not seen, then its register value will be 00. If a TE was found on port 2, the value will be 04,

because port 2 corresponds to bit 2. If a DO was found on port 5, the value will be 20, because port 5 corresponds to bit 5. If you are unfamiliar with such notation, you may find it useful to translate the hexadecimal values to binary first. Then you can find which bit is on and find the corresponding port that matches it: simply count from right to left, starting at 0.

- It is not important to recognize whether the error was a TO, a TE, a DO, so on. This information is left in for historical reasons and may be useful to development, but not to making a FRU call.
- For adapters, indicated as `VDC, id X`, the VDC_FLT and MSTAT are important pieces of information.

Master/Slave Miscompare (Only HiPS)

Knowing whether a master/slave miscompare has occurred between a switch chip pair is critical in making a FRU call on faults reported by switch chips. Recall that each switch chip function is actually performed by two physical chips. Both chips are taking input from the rest of the system. The master chip is providing the outputs to the rest of the system. The slave chip is using the inputs from the rest of the system and checking the outputs from the master to see if they make sense. When they do not make sense, an error is flagged. Although this is an expensive way of checking for errors, it is quite thorough, because in order to let an error escape, both chips would have to break in the same way at the same time.

In the flt file, the master is noted as “chip 0” and the slave is “chip 1.” For example, `Switch, id 10036, chip 0 time = 006ef91f5a65` is a report from master chip 10036, while `Switch, id 10036, chip 0 & 1 time=006ef91fa65` indicates that both the master and slave chip saw the same thing.

If the master and slave report different errors, then this is known as a *master/slave miscompare*. This indicates that they each saw something different. If a particular port is reported in error, by either the master or slave, then there is a good chance that the problem is either with the board or the cable. The cable may be the problem, because it may be attenuating the signal such that it reaches a voltage in no-man′s land and one may decide it is a zero and the other may decide that it is a one, and there is your master/slave miscompare.

If the chips cannot narrow things down to a port, the odds increase that the problem is with either the master or slave switch chip.

If the master and slave report the same error, then it is more likely that the problem is on the other side of the link/cable, or it is the cable.

If the port called is not on board (port 0 through 3), you should always check the cable and connectors for bent pins.

If an onboard port (port 4 through 7) is called, the FRU is the switch assembly that houses the chip that is reporting the fault.

4.7.5 Fault Syndrome on the flt File

The fault syndrome register is the register that collects error information in the adapter. There are two classes of error with which we should be concerned:

- VDC errors occur on the link. There are bits 0, 4 through 11 and 13. These could be the adapter, the cable, or the switch port to which the adapter is connected. Bit 8 can also be forced by software/microcode when it finds a problem that it feels may cause a switch fault. True VDC errors are indicated by an MSTAT equal to 000000c0.
- Adapter/node errors are indicated by the other bits. These indicate either a problem with the adapter or the node.

In all cases, diagnostics should be run.

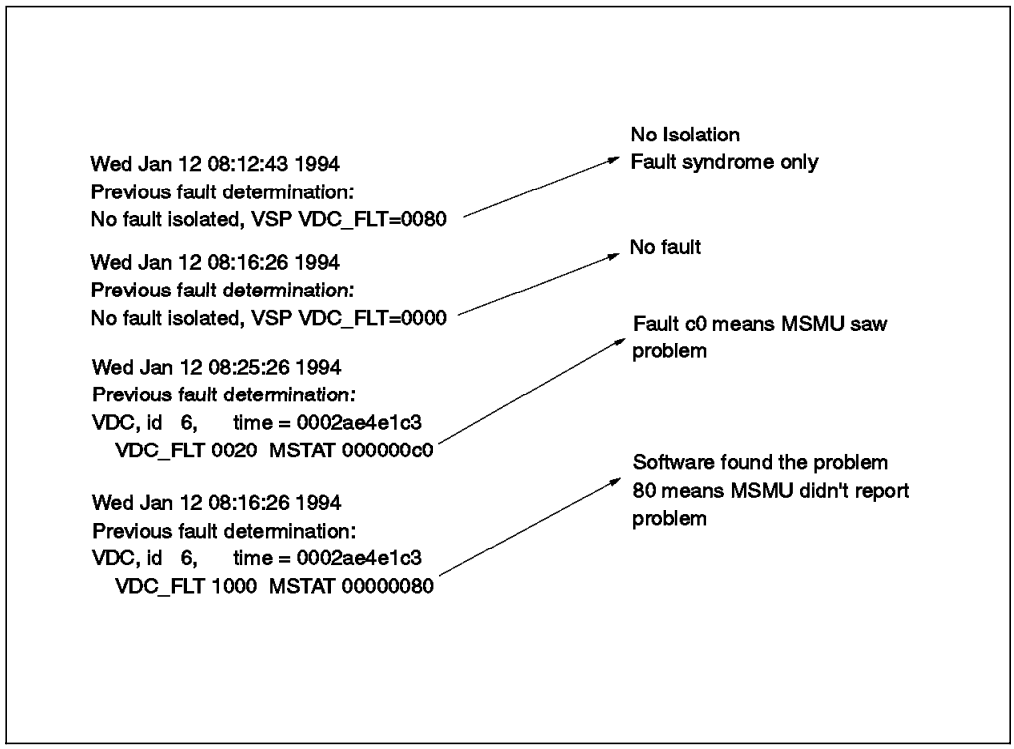


Figure 56. Fault Syndrome on flt File

MSMU Status Register

The MSMU status register is almost always:

- 00000080 - which is normal
- 000000c0 - which indicates a link error
- 12345678 - which indicates that software expected a fault and the information in the fault record is to be ignored.

Any other status reported probably means that the adapter has a problem. Run diagnostics.

4.8 Additional Problem Determination

The following sections discuss determining additional system problems.

4.8.1 Debugging Scripts

If a problem could not be determined by the log files, it is useful to debug the command which fails by itself. Several commands used for SP Switchs are shell scripts or perl scripts.

Shell scripts could be debugged by inserting `set -x` in the beginning of a script. It should be inserted in a new line right after `#!/bin/ksh`.

Perl scripts could be debugged by inserting `-d` flag just in the first line of the script.

```
#!/usr/lpp/ssp/perl/bin/perl -d
```

- Run the commands:

  ```
  # script <filename>
  # <name of perl script>
  ```

  – From within the perl script enter:

    - ″t″ (to turn trace on)
    - ″c″ (to continue)

  – Then run:

    ```
    # exit      (to stop the script command)
    ```

4.8.2 AIX Error Logging

In addition to the log files, there is always a possibility of errors being recorded in the error log (/var/adm/ras/errlog). These errors can be read by using the error report command `errpt`. It is important to note the time of the failure in order to correlate messages in error report `errpt` to the error. There are also many normal, informational switch messages that you will see in the error report. For instance, when you issue an `Estart`, you will see a switch fault in the error report.

Expect one of each of these messages for each `Estart` command.

```
ERROR_ID  TIMESTAMP T CL Res Name   ERROR_Description
34FFBE83 0502140496 T H  Worm       HPS Fault - detected by switch chip
C3189234 0502135796 T H  Worm       HPS Fault - not isolated
```

Figure 57. Message in Error Report while Estart Has Been Executed

To get some error reports specific to the switch, you can use `dsh -a errpt -a -N css` or `dsh -a errpt -a -N Worm`. You can also run diags if you suspect a hardware problem on the switch: `diag -c -d css0` or `diag -A -d css0`.

Chapter 5. System Partitioning

This chapter describes System Partitioning in a way that will enable the reader to understand how all the key components fit together, and give an understanding of how to carry out problem determination based on this information.

The way in which the RS/6000 SP can be partitioned is dependent on the topology of the High Performance Switch, and so this is where the topic is started after a brief introduction. Even in those instances where the High Performance Switch is not installed, System Partitioning follows the topology of what it would expect to see if a High Performance Switch was installed in each frame. It is essential that the reader fully understands the information laid out in the previous chapter on the High Performance Switch before beginning to read the following details about System Partitioning.

5.1 Scope, Rules, and Limitations

It is important to understand which components are affected by System Partitioning before carrying out any of the associated procedures. In this way you can prevent many of the problems that you might otherwise encounter.

In addition, it may be of assistance in deciding whether System Partitioning is suitable for the environment or what is the most suitable configuration to choose.

5.1.1 Scope of System Partitioning

The following components of the POWERparallel System Support Programs are affected by System Partitioning:

- High Performance Switch
- The Heartbeat
- System Data Repository (SDR)
- Job Manager

These topics will be dealt with individually in far greater detail later in the chapter, but it is worth noting that all these components maintain a completely separate and non-interfering environment in each partition that you set up. The reorganization of these subsystems is largely managed by the scripts that are provided in POWERparallel System Support Programs Version 2.1 that can be run from the SMIT screens for System Partitioning. You can use the following SMIT fastpath, for example:

```
# smit syspar
```

Many components remain systemwide and are not affected by System Partitioning. These components continue to be administered in exactly the same manner both before and after setting up the partitions. Some examples are:

- Kerberos
- User Management
- Accounting

5.1.2 Some Rules for Partitioning

There are some basic rules that should always be followed when you set up the partitions which impose some limitations on the configurations that can be selected for the partitions.

All nodes on a switch chip must be in the same partition. The previous chapter outlines which switch chips service which nodes. From this, it can be seen that in a single frame, single switch system, with two wide nodes in slots one and three, it would not be possible to have these two nodes in a partition on their own. Later in the chapter we will look at some further configurations in more detail.

In a POWERparallel System Support Programs Version 2.1 system where you have not explicitly set up System Partitioning, you still have a single, default, partition. Many of the configuration files and variables are in place; it is just that they always point to the only partition that exists.

Within a partition, it is essential to have all the nodes installed with the same version of AIX and POWERparallel System Support Programs. If by some chance there are mixed versions within a partition, then many components will fail to work properly (if at all), inconsistencies will appear in some of your output, and the configuration will not be supported. In this case, these nodes should be reinstalled at the correct level to ensure consistency throughout the partition. Check the versions of products installed, using either the `lslpp` command or the `oslevel` command. For example, the following command can be run to check the version of POWERparallel System Support Programs:

```
# lslpp -l | grep ssp
```

If there is a requirement to have more than a single AIX Version 3.2.5 partition, then ensure that a recent level of maintenance is applied. Check the level by issuing the following command:

```
# lslpp -L | grep ssp.basic
```

Here is an example of the output from this command:

```
ssp.basic           2.1.0.10   C     SP System Support Package
```

To provide the support for multiple AIX Version 3.2.5 partitions, ensure that the level of *ssp.basic* is at least 2.1.0.8. An AIX Version 3.2.5 partition is always the *default* partition, but this topic will be addressed in more detail later on.

The Control Workstation must be at AIX Version 4.1 and PSSP 2.1 to provide the support for System Partitioning. All the additional High Performance Switch topology files required for the many partitioning configurations are supplied in the product *ssp.top*. Ensure this product is installed on the Control Workstation before beginning any tasks associated with System Partitioning.

When the Control Workstation is installed at PSSP 2.1 and AIX Version 4.1, it is only able to act as an install server for nodes at the same version of code. It is not possible to install PSSP 1.2 and AIX Version 3.2.5 nodes from this Control Workstation. It is essential, therefore, to create at least two boot/install servers on the AIX Version 3.2.5 nodes prior to upgrading your Control Workstation to AIX Version 4.1. More details on how to do this are in Chapter 2, “The Installation Process” on page 7.

The reason that at least two boot/install servers are needed is so that if one of the boot/install servers needs installing, there is another node from which to

carry out this task. The underlying reason behind this is that PSSP 2.1 uses a product called the Network Installation Manager (NIM) to carry out many of the tasks associated with the installation of the nodes. NIM is not compatible with AIX Version 3.2.5.

While creating partitions with different versions of AIX, be extremely careful when rebooting the nodes after applying the System Partitioning, and ensure that the desired action has taken place. If the node should network install, then check that this has actually happened. Similarly, if the node should be rebooting from disk to the same version of AIX, then ensure this also actually happened. If any discrepancies are found, then attempt to rectify the situation immediately.

5.1.3 Some Limitations of System Partitioning

Before beginning System Partitioning, it is a good idea to assess whether the following limitations will have an effect on the configurations that can be applied:

- Shared hard disk drives must be connected to nodes in the same partition.
- HACMP (Highly Available Clustered Multi Processors) nodes must be in the same partition.
- Virtual Shared Disks (VSDs) cannot span partitions.

If there are different versions of AIX in the partitions, then it is obvious to see how the issue of compatibility of code makes these limitations essential. Even when all the partitions are running the same versions of code, there is still good justification for these limitations.

Typically, System Partitioning is set up to logically separate applications (for example, to separate applications or company departments). In these instances, there would be little sense in having a node take over the resources of another node in a different partition using HACMP.

Similarly, there would be little point in spreading a database across two partitions using VSDs, when the partitions service completely different departments that do not use the same database tables. If these two departments did use the same database tables, then the database must reside within one partition when using VSDs.

5.2 Partitioning and the High Performance Switch

The High Performance Switch is central to the configuration of System Partitioning. Even in systems that do not have a High Performance Switch installed, the configuration is still based on the assumption that a switch is installed in each frame. In other words, similar sets of rules apply to configuring the System Partitioning, regardless of whether a High Performance Switch is installed or not. In fact, the only difference is for systems that have a High Performance Switch that services nodes in more than one frame, but this issue will be addressed in more detail later in this chapter.

5.2.1 Partitioning a Single Switch/Frame System

Figure 59 on page 131 shows a single switch board and the four switch chips that service the nodes. On the basis that all nodes on a single switch chip must be in the same partition, the maximum number of partitions that are possible is four. The minimum number is the default of one. All combinations in between are also supported, as shown below:

Nodes	Description
16	The default of a single partition of all 16 nodes
4_12	4 nodes in one partition and 12 nodes in another
4_4_8	2 partitions of 4 nodes and one of 8 nodes
4_4_4_4	4 partitions of 4 nodes
8_8	2 partitions of 8 nodes

For the 16 and 4_4_4_4 configurations, there is only one possible layout for each one. However, for the other configurations it is possible to choose from different layouts, grouping different combinations of nodes in different partitions while still observing the rule that all nodes on a switch chip must be in the same partition. Taking the example of the 4_12 configuration, there are 4 possible layouts. The 4 node partition can consist of the following node combinations (with all the other nodes in the 12 node partition):

Nodes 1, 2, 5, and 6
Nodes 3, 4, 7, and 8
Nodes 9, 10, 13, and 14
Nodes 11, 12, 15, and 16

This is based on the assumption that all of these are thin nodes. The reality is often that you have a combination of thin and wide nodes, and that the frame may have empty slots. The same rules apply though, and you should consider these groupings as node slots rather than actual nodes. Here is an example of applying a 4_12 configuration where there are 2 wide nodes occupying the bottom 2 drawers, there are 10 thin nodes, the top drawer is empty, and the layout for the 4 node partition is chosen to contain node slots 1, 2, 5, and 6:

4 partition Nodes 1, 5, and 6
12 partition Nodes 3, 7, 8, 9, 10, 11, 12, 13, and 14

The 4 partition does not contain node 2 because this slot is occupied by wide node 1. Similarly, node 4 is absent from the 12 partition. Nodes 15 and 16 are absent because these slots are empty since the top drawer of the frame is unused at present. Therefore the reality is that the 4 partition contains 3 nodes (1 wide, 2 thin) and the 12 partition contains 9 nodes (1 wide, 8 thin).

Even when using the notion of node slots rather than nodes, it is still not possible to accurately represent the situation where you have a single switch servicing more than one frame. This is still a single switch system and the same rules apply that all configurations are supported, as just described, but a closer look at the switch topology is required to assess which nodes will be in which partition in each of the possible layouts.

The following example is based on having 1 switch board servicing 2 frames that are both fully populated with 8 wide nodes. There are a maximum of 16 node connections on a switch board, and in this example they are all utilized.

Following is an insert from the switch topology file for this configuration that shows how the nodes are connected to the High Performance Switch:

```
# Node connections in frame L01 to switch 1 in L01
s 15 3  tb0 0 0     L01-S00-BH-J18 to L01-N1
s 15 2  tb0 1 0     L01-S00-BH-J16 to L01-N3
s 16 0  tb0 2 0     L01-S00-BH-J20 to L01-N5
s 16 1  tb0 3 0     L01-S00-BH-J22 to L01-N7
s 15 1  tb0 4 0     L01-S00-BH-J14 to L01-N9
s 15 0  tb0 5 0     L01-S00-BH-J12 to L01-N11
s 16 2  tb0 6 0     L01-S00-BH-J24 to L01-N13
s 16 3  tb0 7 0     L01-S00-BH-J26 to L01-N15
s 14 3  tb0 8 0     L01-S00-BH-J10 to L01-N17
s 14 2  tb0 9 0     L01-S00-BH-J8  to L01-N19
s 17 0  tb0 10 0    L01-S00-BH-J28 to L01-N21
s 17 1  tb0 11 0    L01-S00-BH-J30 to L01-N23
s 14 1  tb0 12 0    L01-S00-BH-J6  to L01-N25
s 14 0  tb0 13 0    L01-S00-BH-J4  to L01-N27
s 17 2  tb0 14 0    L01-S00-BH-J32 to L01-N29
s 17 3  tb0 15 0    L01-S00-BH-J34 to L01-N31
```

Figure 58. Topology File Insert from a Single Switch 16 Wide Node System

Note that the node numbering does not follow the usual convention. For example, where you would normally expect to see “L01-N2,” there is “L01-N3.” The node numbering in relation to the slot numbers remains consistent, but the physical cabling has changed to enable the connection of 16 wide nodes to the same switch board. The J16 cable now goes to slot 3 (node 3) instead of slot 2 (node 2). This pattern continues up through the 2 frames.

This has an effect on how the nodes are allocated in the various layouts. Using the 4_12 example again, here is one of the possible configurations:

4 partition Nodes 1, 3, 9, and 11
12 partition Nodes 5, 7, 13, 15, 17, 19, 21, 23, 25, 27, 29, and 31

Using the topology files in this way is the only accurate method that can be used to assess the impact of System Partitioning on all possible RS/6000 SP configurations. However, using the *slot number* method is perfectly valid for configurations where a single switch services nodes in a single frame only.

If there was no High Performance Switch in this example of a 2 frame, 16 wide node system, then the partitioning is done with the expectation that there would be a High Performance Switch in each of the 2 frames, and so you would use the *slot number* method to assess the impact of the various layouts on your configuration.

5.2.2 Partitioning a Two Switch/Frame System

A maximum of two partitions only are supported in these configurations. However, any combination is supported giving the following options:

Nodes	Description
32	The default single partition
4_28	4 nodes in one partition, 28 in the other
8_24	8 nodes in one partition, 24 in the other
12_20	12 nodes in one partition, 20 in the other
16_16	16 nodes in each partition

Within all except the 32 configuration, there are many layouts possible. For example, in the 4_28 configuration, there are 8 possible layouts. As there are 8 switch chips servicing the nodes, it is possible to choose any one of these to be in the 4 partition.

In the 16_16 configuration, it is possible to put each frame in a separate partition, or half of each frame in each partition, or any other combination that falls within the basic rules.

Even with the limitation of a maximum of two partitions, there are a large number of permutations possible. This accounts for the large number of files present in the *ssp.top* product and explains why these limitations were imposed to prevent the product becoming enormous in size. It is possible to obtain topology files that will support a configuration that is not shipped as standard in ssp.top, by making a special request (RPQ)[2].

5.2.3 Partitioning a Three or More Switch/Frame System

In this configuration, a maximum of three partitions is allowed and all nodes within a frame (or serviced by a single switch board) must be in the same partition.

For example, in a four switch/frame system, the following are supported configurations:

Nodes	Description
64	Default single partition system
16_48	1 frame in a single partition, 3 frames in the other
32_32	2 frames in each partition
16_16_32	2 partitions of 1 frame and another of 2 frames

Once again, there are many layouts for all but the default single partition.

[2] Contact your IBM representative for further details.

5.2.4 Switch Topologies

The effect of partitioning the High Performance Switch is that there is no interference between the partitions that have been set up. Anything that happens in one partition will have no effect on the others. If a switch fault is generated in one partition, this will not affect the switch in the others. The fabric of the other partitions will remain stable while the Worm daemon on the primary node in the partition that took the switch fault will attempt to recover the switch network in that partition. This behavior is ensured by each partition having a different topology file that does not contain any routes that would allow traffic to flow between partitions. As a result, nodes in one partition have no knowledge of how to communicate with the nodes outside its own partition. The Worm daemon on the primary node is responsible for generating the routing tables, which it then distributes to the switch adapters on the nodes for storing in their NVRAM.

Each partition has its own primary node, so that running `Estart` in one partition has no effect on the others. The Worm on the primary can carry out its tasks in a completely independent manner from the other partitions.

Generating the correct routes for the partition based on the topology files that get selected during the *apply* of the System Partitioning is critical to the health of the switch in that partition.

Following is an example of how a single switch system with 16 thin nodes could be partitioned using one of the layouts for the 8_8 configurations.

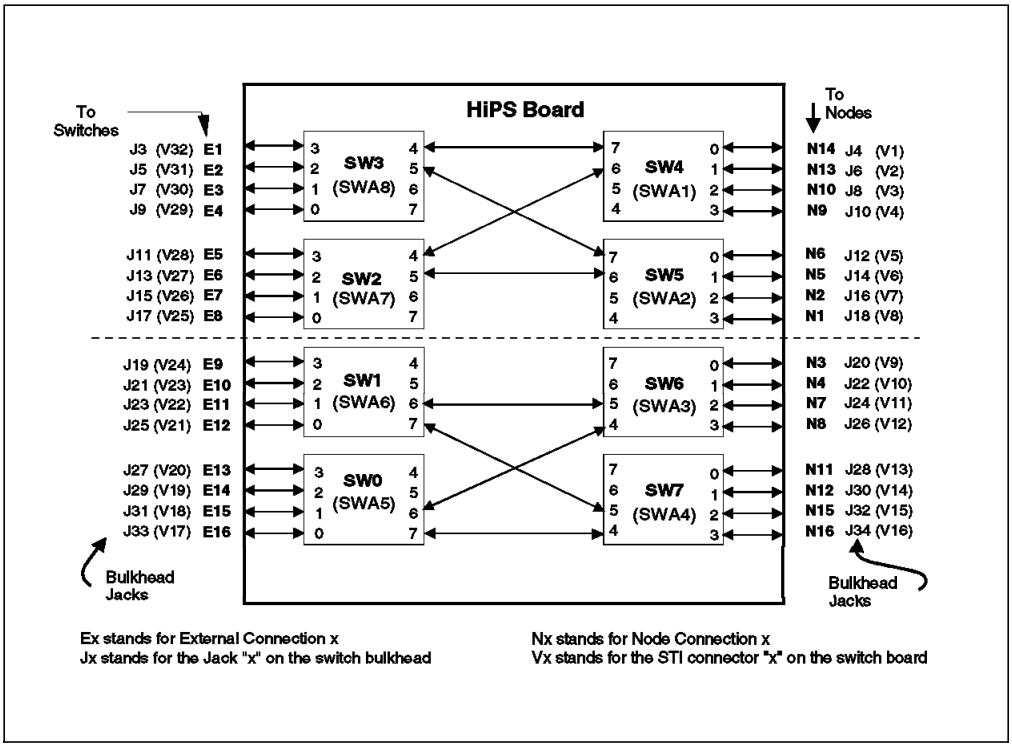


Figure 59. Example of an 8_8 Configuration

By deleting half of the routes (8 of the 16), there is no communication possible between the two partitions. However, within the 8-way partition, there are only two possible routes between each and every node (where previously there were four). Halving the bandwidth in this way does have an impact on performance,

but not by 50%. Testing has shown that the expected performance reduction is usually within the 10 to 15% range. Much will depend on the specifics of the particular system, and the type of network traffic generated on the switch.

With the High Performance Switch, the IP protocol only uses one path at a time[3] , so the impact is likely to be less. There is still a scope of packet contention in a busy switch network, because if a path is blocked, then only one alternative exists. The user space message passing mode protocol is able to use multiple paths, and so the performance impact may be greater depending on the type of network traffic.

This protocol is used by parallel programs that require a high bandwidth and low latency for their message passing. In addition, the impact of losing one of the internal routes on the switch board becomes far more severe in the 8-way partition, because only one path will be available for communicating between nodes on different switch chips. The chance of experiencing packet contention is increased considerably, which could have a major impact on performance. In an unpartitioned system, the effect of losing an internal link or route would be minimal and the switch could run indefinitely until an opportunity is available to carry out maintenance.

Similar considerations can be applied to the 12-way partition, but the impact is minimal because 3 out of the 4 routes are still available.

Based on the 8_8 example previously given, the topology files shown in Figure 60 illustrate how the routes have been deleted.

```
# Node connections in frame L01 to switch 1 in L01
s 15 3  tb0 0 0          E01-S17-BH-J18 to E01-N1
s 15 2  tb0 1 0          E01-S17-BH-J16 to E01-N2
s 15 1  tb0 4 0          E01-S17-BH-J14 to E01-N5
s 15 0  tb0 5 0          E01-S17-BH-J12 to E01-N6
s 14 3  tb0 8 0          E01-S17-BH-J10 to E01-N9
s 14 2  tb0 9 0          E01-S17-BH-J8 to Exx-Nxx
s 14 1  tb0 12 0         E01-S17-BH-J6 to E01-N13
s 14 0  tb0 13 0         E01-S17-BH-J4 to Exx-Nxx
# On board connections between switch chips on switch 1 in Frame L01
s 14 7    s 13 4         E01-S17-SC
s 14 6    s 12 4         E01-S17-SC
s 15 7    s 13 5         E01-S17-SC
s 15 6    s 12 5         E01-S17-SC
```

Figure 60. Insert of Topology File for the First 8-Way Partition

Node connections are present only for the nodes in the partition, and there are only 4 links present for the onboard switch chip connections. When diagnosing configuration problems, check that the correct topology file is being used for the partition you are having difficulty with. The topology files are stored in the SDR directory:

`/spdata/sys1/sdr/partitions/<IP alias address>/files`

[3] With the new SP Switch this is no longer the case; IP can use multiple paths.

IP address aliasing will be dealt with in detail later in the chapter, but it is sufficient to know at this stage that each partition is identified by having a different alias. The topology file for the second partition is shown in Figure 61 on page 133 for completeness.

```
# Node connections in frame L01 to switch 1 in L01
s 16 0  tb0 2 0          E01-S17-BH-J20 to E01-N3
s 16 1  tb0 3 0          E01-S17-BH-J22 to E01-N4
s 16 2  tb0 6 0          E01-S17-BH-J24 to E01-N7
s 16 3  tb0 7 0          E01-S17-BH-J26 to E01-N8
s 17 0  tb0 10 0         E01-S17-BH-J28 to E01-N11
s 17 1  tb0 11 0         E01-S17-BH-J30 to Exx-Nxx
s 17 2  tb0 14 0         E01-S17-BH-J32 to E01-N15
s 17 3  tb0 15 0         E01-S17-BH-J34 to Exx-Nxx
# On board connections between switch chips on switch 1 in Frame L01
s 16 5    s 11 6         E01-S17-SC
s 16 4    s 10 6         E01-S17-SC
s 17 5    s 11 7         E01-S17-SC
s 17 4    s 10 7         E01-S17-SC
```

Figure 61. Insert of Topology File for the Second Partition

This example is based on the Jack cable (Jxx) numbering system used on the High Performance Switch, rather than on the newer SP Switch. The example is based also on a system with eight thin nodes (1-8) and four thin nodes (nodes 9, 11, 13, and 15). The right side of the topology file shows “Nxx” to represent where a wrap plug is inserted (rather than a Jack Cable that would normally be present to connect to a node). In this case, the wrap plug is required due to a wide node being present in that particular drawer.

There is another issue that should be considered in relation to performance when setting up System Partitioning on a multi-switch system. Take an example of a *5nsb.0isb* configuration that is not currently partitioned. If you have a resource pool of 20 nodes set up, for example, for running parallel jobs across the switch, then the best placement of the nodes is likely to be with four nodes on each switch to provide the greatest bandwidth possible. In this case, where you are not using Intermediate Switch Boards (ISBs), there are only four cables available to connect to each of the other switch boards, given that there are only 16 external Jack sockets available for external connections (rather than node connections) and there are four other switches to connect to.

By placing the nodes evenly across the switches, each of the four nodes can communicate with any of the other four nodes over four data cables. If you placed ten nodes each on two switches only, then these ten nodes would only be able to communicate over four cables to the other ten nodes and the bandwidth is reduced considerably.

If a large amount of data is flowing across the switch during these parallel jobs, then the chance of packet contention over the routes is increased by concentrating the nodes in the resource pool on fewer switches. Of course, if you can place all the nodes on one switch, you will resolve the issue; but in this case, it is not possible because the pool consists of 20 nodes.

It is worth taking this issue of *bisectional bandwidth* into account when partitioning the system, and when you have this type of resource pool where

performance across the switch is of great importance. By partitioning the system, the nodes may have to be concentrated on the switches, therefore reducing bandwidth.

5.2.5 Configuration Files

All the switch configuration files at PSSP 2.1 are in the following directory:

```
/spdata/sys1/syspar_configs
```

The system size directories are below this directory based on the number of switch boards in the system (or frames, if there is no switch). For example:

```
/spdata/sys1/syspar_configs/1nsb0isb
```

Beneath these directories are the config directories which determine how many partitions will be in the system and how many nodes will be in each one. For example:

```
/spdata/sys1/syspar_configs/1nsb0isb/config.8_8
```

Beneath this are the layout directories which determine which nodes will go into which partition. For example:

```
/spdata/sys1/syspar_configs/1nsb0isb/config.8_8/layout.1
```

There is a file in this directory called *layout.desc* that describes the actual layout in terms of which node goes in which partition. Beneath this are the syspar directories, one for each partition that has been selected from the config directory. For example:

```
/spdata/sys1/syspar_configs/1nsb0isb/config.8_8/layout.1/syspar.1
/spdata/sys1/syspar_configs/1nsb0isb/config.8_8/layout.1/syspar.2
```

Within these directories there are three files: topology, nodelist and custom. Actually, of these four files (these three plus the layout.desc), only the custom file is a real file; all the others are symbolic links to files in the following directories:

```
/spdata/sys1/syspar_configs/topologies
/spdata/sys1/syspar_configs/descriptions
/spdata/sys1/syspar_configs/nodelists
```

The topology file we have already discussed earlier in the chapter. The nodelist file is similar to the layout.desc file, except that it has been split out into its respective partition and contains only a list of its own nodes.

The custom file is the only one that gets modified. The others are selected as supplied to match the requirements of your partitioning. The custom file gets updated during the *apply* of the configuration, and this will be covered later in more detail. It will contain information about the version of POWERparallel System Support Programs in the partition, for example.

The diagram in Figure 62 on page 135 is intended to assist in the understanding of the directory structure, although you only really need to use this information if you need to debug problems with System Partitioning or the switch. All the configurations can be done from the SMIT panels without this knowledge.

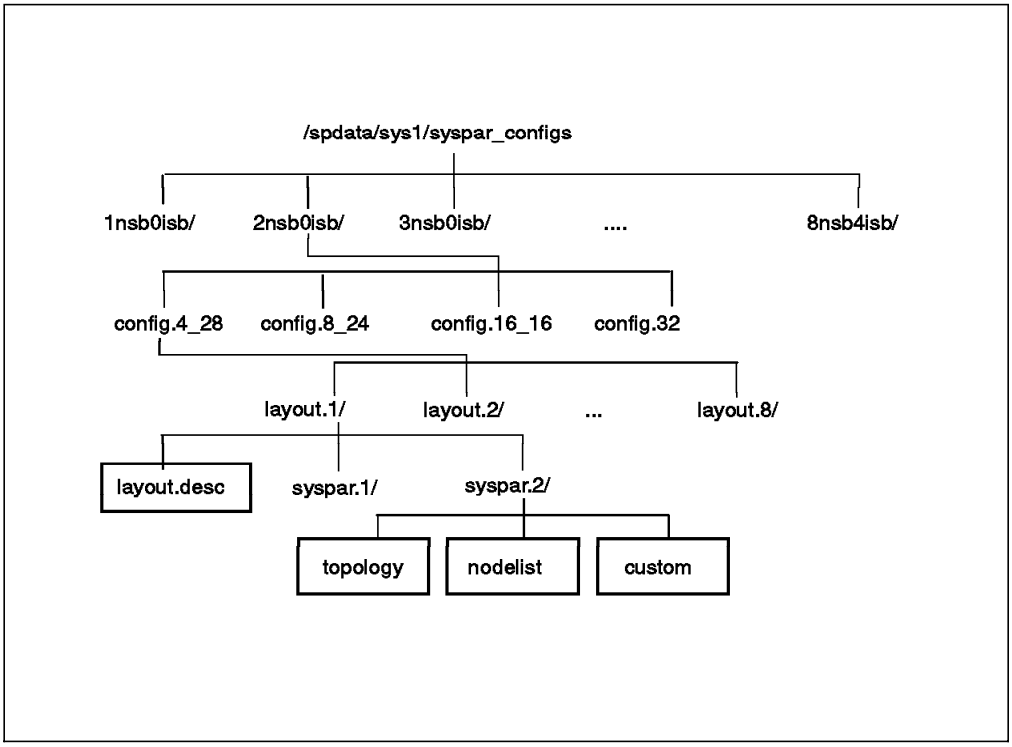


Figure 62. Directory Structure for the Topology Files in PSSP 2.1

Returning to the example of an 8_8 configuration, following is a list of all the associated topology files:

```
/spdata/sys1/syspar_configs/topologies/any.l1.8way1.0isb
/spdata/sys1/syspar_configs/topologies/any.l1.8way2.0isb
/spdata/sys1/syspar_configs/topologies/any.l1.8way3.0isb
/spdata/sys1/syspar_configs/topologies/any.l1.8way4.0isb
/spdata/sys1/syspar_configs/topologies/any.l1.8way5.0isb
/spdata/sys1/syspar_configs/topologies/any.l1.8way6.0isb
/spdata/sys1/syspar_configs/topologies/any.l2.8way1.0isb
/spdata/sys1/syspar_configs/topologies/any.l2.8way2.0isb
/spdata/sys1/syspar_configs/topologies/any.l2.8way3.0isb
/spdata/sys1/syspar_configs/topologies/any.l2.8way4.0isb
/spdata/sys1/syspar_configs/topologies/any.l2.8way5.0isb
/spdata/sys1/syspar_configs/topologies/any.l2.8way6.0isb
```

Note that there are 12 files, 6 beginning with *any.l1*, and 6 with *any.l2*. As there are three possible layouts for an 8_8 configuration, this explains why there are six of each because you have one topology file for each of the partitions.

In the supported configurations, it is possible to have an 8-way partition only in a single switch/frame system or in a two switch/frame system, because in a three or more frame system, all the nodes serviced by a switch or in a frame must be in the same partition.

The *l1* or *l2* part of the filename refers to *logical frame* and *logical frame 2*, respectively. The High Performance Switch has this concept of “logical frame” because it is possible to have one switch servicing two physical frames. The 6 *l1* files therefore refer to an 8-way partition in the first logical frame and the 6 *l2* files similarly refer to one in a second logical frame. This is important because within the topology file, the second digit refers to the logical frame number and

this information is essential during Estart if the Worm on the primary node is to successfully initialize the switch in that partition.

In cases where the switch will not initialize within a partition, check that the topology file was copied correctly into the files directory of the SDR by running the following commands:

```
# cd /spdata/sys1/sdr/partitions/<IP alias>/files
# ls -l
```

The correct file should be listed in this directory. Take a look at the contents to check that it contains the correct topology.

In addition, check that the SDR object class *Switch_partition* contains the correct filename for the topology file that is loaded in the SDR files directory. To do this, run the following commands:

```
# export SP_NAME=<syspar_name>
# SDRGetObjects Switch_partition topology_filename
```

The SP_NAME variable can be used to set the current partition. If this variable is not set, then the current partition will be the *primary* partition as specified in the node's /etc/SDR_dest_info file. Commands such as SDRGetObjects will only provide information about the current partition. This topic will be dealt with in more detail later in the chapter. Any discrepancy in the two filenames will cause Estart to fail.

In this example of an 8_8 configuration, using layout.1, the two files that will be loaded into the SDR files directories for each partition should be:

- /spdata/sys1/syspar_configs/topologies/any.l1.8way1.0isb
- /spdata/sys1/syspar_configs/topologies/any.l1.8way2.0isb

Run the following commands to check that SDR contains the correct information about the topology files for each partition:

```
# export SP_NAME=<IP alias>
# SDRGetObjects Switch_partition topology_filename
```

The output of this SDRGetObjects command should be:

```
topology_filename
any.l1.8way1.0isb
```

For the other partition, it should be:

```
topology_filename
any.l1.8way2.0isb
```

It does not matter that you have multiple topology files in the files directory of the SDR (although it may be preferable to remove redundant ones), as long as the *Switch_partition* object class is pointing to the correct one.

Debugging the switch on a partitioned system is similar to that on an unpartitioned system. The specific differences will be covered throughout this chapter, but the fundamentals remain the same. To recap briefly some of the actions that may be appropriate in these circumstances:

- Run errpt on all nodes to look for switch or Worm errors.
- Look at the log files in /var/adm/SPlogs/css on the primary node in the partition.
- Ensure /etc/SP exists on the primary node in the partition.

- Ensure the file /etc/SP/expected.top does not exist on the primary node in the partition.
- Check that the Worm daemon is running on the primary node in the partition and on all the other nodes.

For more details on these and other appropriate debugging options, refer to Chapter 4, “The Switch” on page 87.

5.3 Creating Partitions

Once you have decided how to partition the system, all you need to do is to *apply* the chosen configuration. There are some prerequisites, however, that should be considered.

5.3.1 Prerequisites to System Partitioning

Always ensure there are at least two boot/install servers set up for AIX Version 3.2.5 nodes prior to upgrading or reinstalling the Control Workstation to AIX Version 4.1. (Of course, if there is to be no AIX Version 3.2.5 partition, then this simplifies things somewhat.)

After the migration or reinstallation of the Control Workstation, verify that all the required products are installed by running:

```
# lslpp -l | grep ssp
```

In particular, ensure that *ssp.top* is installed to provide the topology files for System Partitioning. In addition, ensure that the latest maintenance is applied to the POWERparallel System Support Programs products to avoid the added complication of encountering any known problems[4].

If the system was previously installed at AIX Version 3.2.5, it is a good idea to run with the AIX Version 4.1 Control Workstation for a period of time to ensure that the installation was successful for all components of the POWERparallel System Support Programs. The AIX Version 3.2.5 nodes can run in production mode, and there will then be confidence that the installation was completely successful.

After this period, prepare the nodes that are to run AIX Version 4.1 for installation by running setup_server[5] . Further detail about setup_server script can be found in Appendix A, “RS/6000 SP Script Files” on page 229 under A.3, “The setup_server Script” on page 239.

If all the nodes are to be installed at AIX Version 4.1, then the installation can proceed normally and the partitioning can be set up as required when the time is appropriate. However, if there are to be mixed partitions (that is, both AIX Version 4.1 and AIX Version 3.2.5 partitions), then the System Partitioning must be *applied* before the AIX Version 4.1 nodes are installed. The current situation is that mixed nodes in a partition are not supported because there are a number of problems associated with having this type of environment.

---

4 Contact your IBM Support Center to obtain guidance on the PTFs you should install.

5 Refer to the *SP Installation Guide*, GC23-3898 for details on this procedure.

Prepare for System Partitioning by setting up an IP alias for each additional partition that is to be created. The IP alias function is provided in the standard implementation of TCP/IP in AIX. It allows the association of more than one IP address with a network interface. For partitioning to work, it is essential that an alias is created for the interface that is associated with the hostname of the Control Workstation. To find out the hostname, use the `hostname` command on the Control Workstation. To find out which interface is associated with the Control Workstation hostname, run the following command:

```
# netstat -r | grep <hostname of CWS>
```

The output of this command will show which interface to create the alias names for. Typically the interface will either be en0, tr0, or fi0. Following is an example of how to create an alias for the *en0* network interface:

```
# ifconfig en0 alias 9.180.6.198 netmask 255.255.255.0 up
```

Check that the IP alias became associated with the adapter by running the `ping` command. For example:

```
# ping 9.180.6.198
```

Ensure that the IP address used is not already in use by another system on the network. In addition, ensure that there will not be a conflict in the future should you add any additional nodes. Choose an IP address that will be definitely out of the range even if you upgrade your RS/6000 SP by adding a large number of nodes.

The next step is to ensure that the IP alias that has just been set up can be resolved by the name resolution mechanism. If you are using the /etc/hosts file for this purpose, then simply add the IP address and an appropriate name. (These hostnames will be used to identify your partitions when carrying out system administration tasks on the RS/6000 SP, so bear this in mind when choosing the names.) Propagate these changes to the nodes either manually by using `ftp`, for example, or by using one of the POWERparallel System Support Programs commands like `supper` or `pcp`.

If you are using the product Network Information Services (NIS), for example, ensure that you have a route set up to get to a NIS server from the IP alias address. Check to see which route is being used for the Control Workstation IP address so that the same route can be used for the IP alias, should this be appropriate. Add the aliases to the NIS maps and ensure all slave servers are updated by running a `yppush`. It is a good idea to have slave servers on the nodes because if name resolution is not quick and efficient many of the administrative tools on the RS/6000 SP will not function properly. Following is an example for adding a route for the alias 9.180.6.198:

```
# route add -net 9.180.3.0 netmask 255.255.255.0 9.180.6.198
```

Check that name resolution is working for the aliases by running:

```
# host <alias name>
```

In addition, ensure that the IP alias is persistent after a system reboot by adding the details of the `ifconfig` described previously, to the rc.net file so that it will run each time from the /etc/inittab.

If name resolution is not working, the following steps will not be successful when the attempt is made to set up System Partitioning. Resolve the network problems at this stage and only then continue setting up the partitions.

5.3.2 Process Overview

If everything is successful so far, the next steps can all be carried out from the SMIT menus by running:

```
# smit syspar
```

A menu will appear, as shown in Figure 63.

```
                          System Partition Configuration

Move cursor to desired item and press Enter.

  Archive System Data Repository
  Select System Partition Configuration
  Display Information for Given Configuration or Layout
  Select System Partition Layout
  Enter Customization Information for a Selected System Partition
  Apply System Partition Configuration
  Restore System Partition Configuration

F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 63. SMIT Menu for the Fastpath "syspar"

Carry out the steps in the order presented in the menu, starting at the top with Archive System Data Repository. The flow chart shown in Figure 64 on page 140 matches these steps.

Verifying the configuration can be done at the menu Apply System Partition Configuration.

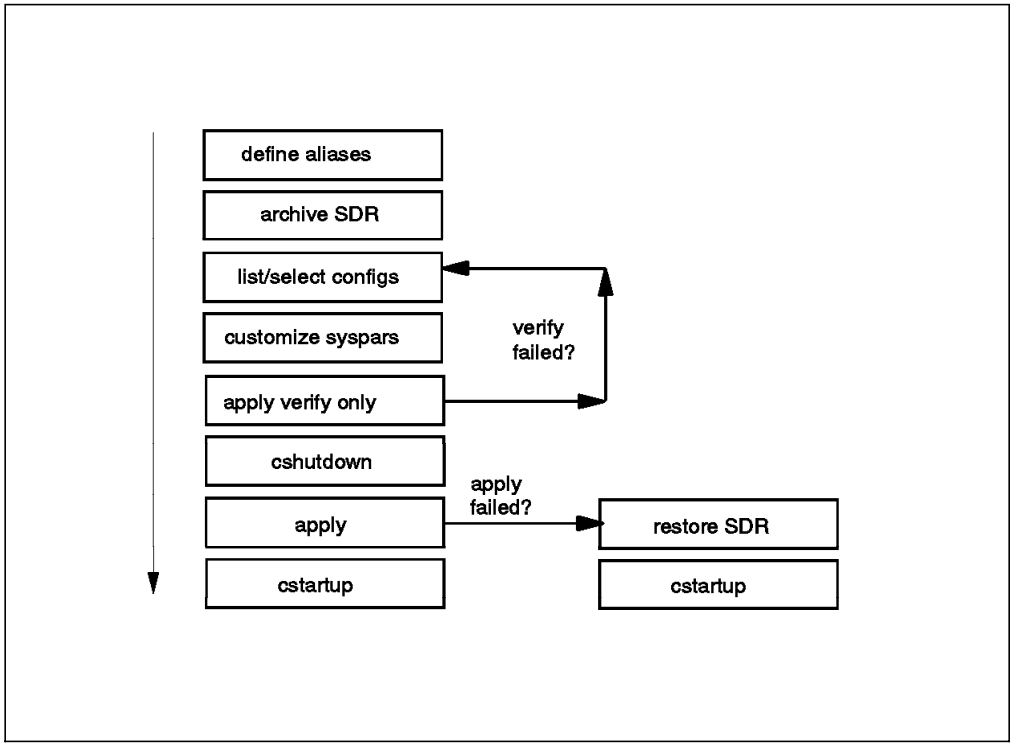


Figure 64. Flow Chart of the System Partitioning Process

The 2nd, 3rd, and 4th steps in the SMIT menu (Select System Partition Configuration, Display Information for Given Configuration or Layout, and Select System Partition Layout) are represented by the single box *list/select configs* in the flow chart. These SMIT menus can be used to assist in deciding how to partition the system, but system partitioning can also be done manually by following the guidelines regarding the directory structure previously given in this chapter.

The `cshutdown` and `cstartup` commands can be run from the `spmon` GUI, or from the command line, when the affected nodes are to be shut down or started up.

5.3.3 Archiving the SDR

Always begin the System Partitioning by archiving the SDR. It is possible to return to the original configuration by *reapplying* it, but this could involve a considerable amount of work and requires detailed knowledge of System Partitioning and the related subsystems. Archiving the SDR can be done from SMIT or the command line. Either way, the command run will be:

```
# /usr/lpp/ssp/bin/SDRArchive
```

Make a note of the name of the archive, because this command may have been run multiple times, leaving several files in the following directory:

```
/spdata/sys1/sdr/archives
```

It should be easy to identify the backup just taken, because the filename is appended with the time. For example:

```
backup.96124.0926
```

After the *backup* part of the name, the *year* is listed, followed by the *Julian day* (that is, the number of days into the year), and then the *time* as the suffix.

5.3.4 Customizing the Partitions

Having selected the configuration (based on the number of switch boards or frames in the system), chosen the layout you require (based on which nodes are required in which partition), and selected that layout, the task of customizing each of the partitions should be carried out. Figure 65 shows the SMIT menu to Enter Customization Information for a selected System Partition:

```
                Enter Customization Arguments for this System Partition

 Type or select values in entry fields.
 Press Enter AFTER making all desired changes.

                                                [Entry Fields]

   System Partition Name or IP Address          [spcws]
   PSSP Code Level                                    PSSP-2.1
   Default Install Image                        [default]
   Primary Node                                 [1]
   Backup Primary Node                          []
   System Partition Path                           config.8_8/layout.1/syspar.1>









F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 65. Example of SMIT Screen for Customizing Layouts

Fill the fields according to the chosen configuration.

The first field System Partition Name or IP Address should contain the hostname or the IP address for the partition that is described in the layout that is chosen for customization. In the preceding example, this is *layout.1*. It is important to know which nodes are going to be in this partition. If you have an AIX Version 3.2.5 partition, this should be associated with the *hostname* of the Control Workstation so that it becomes the *default* partition. All the nodes in the partitions that are not associated with the *hostname* of the Control Workstation (that is, associated with an IP alias) are known as affected nodes. It is these nodes that must be shutdown before applying the System Partitioning.

The second field PSSP Code Level refers to which version of the POWERparallel System Support Programs will be running in the partition. Use the F4 function key to obtain a list, then make the correct choice.

The third field Default Install Image should match the *mksysb* image that has been set up to install the node. This may not be used now if the nodes are not to be reinstalled at this stage, and the details can always be changed at a later date by running the setup_server command.

The fourth field Primary Node should be chosen from the list generated by pressing F4. Each partition will have its own primary node for switch operations.

The next field, Backup Primary Node, may not be present on the SMIT screen. If recent maintenance has been applied to the system, it may be there because this functionally was introduced with the new code for the SP Switch. If this field is present but the High Performance Switch is installed on the system, leave this field blank; otherwise refer to the previous chapter on the switch for recommendations on which node to choose as the *backup primary node*. If you have an SP Switch installed but do not have this menu option, then it is essential that the required maintenance is installed to support this new switch.

The last field refers to the partition that is being customized. This procedure must be completed separately for each of the respective partitions.

When the *OK* message is displayed after Enter has been pressed, the data that had been entered will have been placed in a file called custom. Based on the example above, this will be in the directory:

```
/spdata/sys1/sypar_configs/1nsb0isb/config.8_8/layout.1/sypar.1
```

Here is a custom file based on this example:

```
sypar-name:                 spcws
IP-address:                 9.180.60.199
primary-node:               1
default-install-image:      default
PSSP-code-level:            PSSP-2.1
backup-primary-node:        default
```

Always check after the customization of the partitions that these files were created in their respective *syspar* directories and that the contents are correct. In the example used, this is the custom file for the *default* partition because it is associated with the hostname of the Control Workstation.

5.3.5 Verify the Configuration

Always *verify* the configuration before attempting to *apply* it. This is done at the SMIT menu Apply System Partition Configuration. Figure 66 on page 143 shows an example of this menu:

```
                        Apply System Partition Configuration

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  System Partition Apply Option                          Verify only.         +
  Correct VSD configuration?                             No. Discontinue      +
  System Partition Path                                  config.8_8/layout.1


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 66. Example of SMIT Screen for Verifying the Configuration

Be particularly careful at this menu because the default for the first field System Partition Apply Option is to *apply this configuration*. Toggle to the *Verify only* option using the tab key which will run the spapply_config command with the -v flag and will not actually run the *apply*.

The second field, Correct VSD configuration, can be left at the default *No. Discontinue* unless you want to correct any inconsistencies that may be apparent in the VSD setup in relation to the new partitions. The fundamental issue is often that the VSDs are spanning more than a single partition. Resolve these issues before you begin System Partitioning, because only *non-fatal* errors can be corrected by using this option that passes the -F to the verparvsd command. By choosing *No. Discontinue*, the verparvsd which gets called from spapply_config will exit if any non-fatal errors are encountered.

A common error that is generated by running *verify* details the fact that the nodes are still up. This is merely a warning at this stage to remind you that the *affected* nodes should be shutdown prior to the *apply*. Here is an example of this message:

```
spapply_config: 0022-149 Warning: node: 1 in system partition spcws
has host_responds value of: 1 .
```

The *host_responds* value of 1 indicates that the node is up. A value of 0 indicates that the node has lost connectivity across the en0 interface and could therefore be down.

If the *verify* fails, then rerun the *customize* operation and try again. If it persistently fails, it is likely to be for one of the following reasons:

1. The layout.desc file does not exist in the layout directory.
2. One of the nodelist, custom, or topology files does not exist in one of the syspar directories.
3. The data in the custom file cannot be read, is incorrect, or is inconsistent.

4. The SDR is not up or is not responding.
5. The current data in the SDR that relates to the partitioning has inconsistencies or is incorrect.

Check and correct any problems associated with points 1, 2, and 3. If everything is correct in these areas, then proceed to the next section to find out how to resolve problems with the SDR.

5.3.6 Applying the System Partitioning

The key step at this point is to ensure that all *affected* nodes are shutdown. This is critical to the health of these nodes in the newly-partitioned environment. Shutting them down is the best way to guarantee that all the daemons and files will be configured correctly in the new partitions. Shut the nodes down in the correct manner according to the normal working procedures on the system (that is, in the correct order from the spmon GUI or the command line).

When the `spapply_config` command is run (without the `-v` flag), the following tasks are carried out:

- The SDR is restructured for the partitions and the multiple daemons are created and started.
- The *heartbeat* and *host_responds* are restructured and the multiple daemons are created and started.
- The switch gets configured according to the new partitions (if there is one installed).
- Validates that the VSD configuration is consistent with the partition being set up.

If the *apply* fails, then record the error message that appears on the SMIT screen. In order to debug this further, read the following sections about the SDR and *heartbeat* subsystems.

5.3.7 Validating the Partitions and Restoring the SDR

The option remains to restore the SDR that was archived earlier in the procedure, but it is worth understanding why the *apply* failed in case the same problem occurs the next time you attempt to implement System Partitioning.

To restore the SDR, go to the SMIT menu Restore System Partition Configuration, list the archived files, and select the appropriate backup or archive to restore. This will run the following command:

```
# /usr/lpp/ssp/bin/sprestore_config
```

This script restores the original SDR and reverses the actions carried out in the *apply* of the partitions. For example, it will stop and remove the multiple daemons.

Even if the *apply* was successful, it is still a good idea to run the following command to verify that the partitioning configuration is good in the SDR before the affected nodes are rebooted:

```
# splstdata -p
```

The `spapply_config` command calls the `spverify_config` command to check that the SDR was updated correctly, but it can be run manually from the command line as a final check.

Do not restore the original SDR if the affected nodes have been rebooted and there are mixed partitions running different version of AIX and POWERparallel System Support Programs. This situation is not supported and the effect of running in this type of environment is unpredictable at present for compatibility reasons. Attempting to resolve these issues is beyond the scope of this book.

5.4 SDR Reorganization

The previous section mentioned that SDR was restructured during the *apply* of the System Partitioning. It is important to know how this is done in some detail in order to resolve any outstanding problems that may arise.

5.4.1 SDR Daemons

The SDR contains all the RS/6000 SP-specific configuration information. It is a central repository that only resides on the Control Workstation. Access to this information is essential for all the nodes to function properly. If there are any SDR or SDR error messages, it is a good idea to check that the SDR daemon (sdrd) is running. This can be done by running the following command on the Control Workstation:

```
# ps -ef | grep sdr
```

In the example that has been used so far, the output for the process name from the preceding command should look like:

```
/usr/lpp/ssp/bin/sdrd 9.180.6.199
/usr/lpp/ssp/bin/sdrd 9.180.6.198
```

Alternatively, the following command can be run to check the same information:

```
# lssrc -g sdr
```

Here is an example of the output from this command:

```
Subsystem       Group       PID    Status
 sdr.spcws      sdr         10704  active
 sdr.sppart2    sdr         10962  active
```

The subsystem names are uniquely identified by using the name of the partition, rather than the IP address, as a suffix.

In PSSP 2.1, there is one SDR daemon for each partition. Each daemon is uniquely identified by an IP address. In the case of the *default* partition, this is the IP address associated with the *hostname* of the Control Workstation. In the case of the other partitions, this address is the IP alias that was defined for each partition.

A look at the /etc/services shows that the SDR daemons are listening on TCP port 5712. Each node can connect to this port by specifying the IP address associated with its partition (the IP alias or the *hostname* of the Control Workstation) and making a socket connection. In this way, each SDR daemon is able to communicate only with the nodes in the partition that it associates with.

In PSSP 2.1, the SDR daemons are under SRC master control. The inittab entry should look like the following:

```
sdrd:2:once:/usr/bin/startsrc -g sdr
```

The startsrc uses the -g flag in order to start the multiple daemons. It runs the following script to start up the daemons and ensure that multiple instances of the same daemon do not get created:

```
/usr/lpp/ssp/bin/sdr
```

This script passes the IP address associated with the partitions to the respective daemons.

If there are SDR problems on a particular system, always check that all the respective SDR daemons are running. If not, then run:

```
stopsrc -g sdr
startsrc -g sdr
```

Also ensure that the daemons remain up after restarting them. If they are dying, then check in the following directory for log files to get an indication of why this is happening:

```
/var/adm/SPlogs/sdr
```

These logs are covered in more detail later in the chapter.

The *SP_NAME* variable is not set by default. The user needs to explicilty set and export this variable; otherwise the following file is referenced:

```
/etc/SDR_dest_info
```

In the example used, this file contains the following:

```
default: 9.180.6.199
primary: 9.180.6.198
```

This file was actually taken from one of the affected nodes because the IP address of the primary is the IP alias given to the second partition created (that is, not the default).

Following is the /etc/SDR_dest_info file from a node in the *default* partition:

```
default: 9.180.6.199
primary: 9.180.6.199
```

Both the *default* and the *primary* are set to the IP address associated with the *hostname* of the Control Workstation. The node knows how to communicate with the SDR.

The *default* entry in the file serves two purposes. The first is that whenever the node is rebooted, the rc.sp script that runs from the inittab does the following as part of its function:

- Gets the *default* IP address from the /etc/SDR_dest_info file on that node.
- Contacts the SDR and checks the global object class *Syspar_map* to find out whether the IP address entry in the /etc/SDR_dest_info file for the *primary* is correct.
- If it is not correct, then it will update the SDR_dest_info with the correct entry.

This is a critical step in getting partitioning to work and is one of the reasons why the affected nodes should be shutdown. The reboot ensures that the /etc/SDR_dest_info file is updated with the new IP alias address.

The second reason for having a *default* IP address is that if the partition gets deleted by restoring an archive of the SDR, or by *applying* a different partition,

then the node can still reach the SDR by using this IP address (after failing to connect to the *primary*).

In the example used, if there was an AIX Version 3.2.5 partition, the contents of the /etc/SDR_dest_info file would be:

```
backup:   spcws
opstation: spcws
```

Additional code had to be written to allow more than one AIX Version 3.2.5 partition because the AIX Version 3.2.5 nodes had no concept of System Partitioning. Originally there was no mechanism to allow AIX Version 3.2.5 nodes to be in different partitions. Updating the SDR_dest_info files on affected AIX Version 3.2.5 nodes was an addition that was made.

If there are problems after partitioning the system, particularly in relation to the *host_responds* showing red on the *spmon* GUI for certain nodes, then carry out the following commands for the problematic partition:

```
# export SP_NAME=<syspar name>
# SDRGetObjects Syspar_map
```

Ensure all the /etc/SDR_dest_info files on the problematic nodes have the correct IP address for the *primary* entry (or equivalent *opstation* for AIX Version 3.2.5 nodes) by referring to the output of the SDRGetObjects command previously run. Correct any erroneous entries by using a text editor such as vi and then run the following commands:

```
# stopsrc -g hb
# startsrc -g hb
```

Or, if this is an AIX Version 3.2.5 partition:

```
# ps -ef | grep ccst
# kill <PID of ccst process>
```

If the *ccst* process is not running, then attempt to start it manually by running:

```
/usr/lpp/ssp/bin/hb
```

The last part of this procedure involves stopping and starting *heartbeat* daemons. This is dealt with in more detail in 5.5, “Heartbeat Reorganization” on page 152.

5.4.2 SDR Directory Structure

The SDR directory structure was radically changed at PSSP 2.1 partly to accommodate the requirements of System Partitioning. It is no longer located in /var/sdr on the Control Workstation, as was the case in AIX Version 3.2.5, but instead can be found in:

```
/spdata/sys1/sdr
```

Figure 67 on page 148 shows a diagram that illustrates the first level directory structure below the sdr directory:

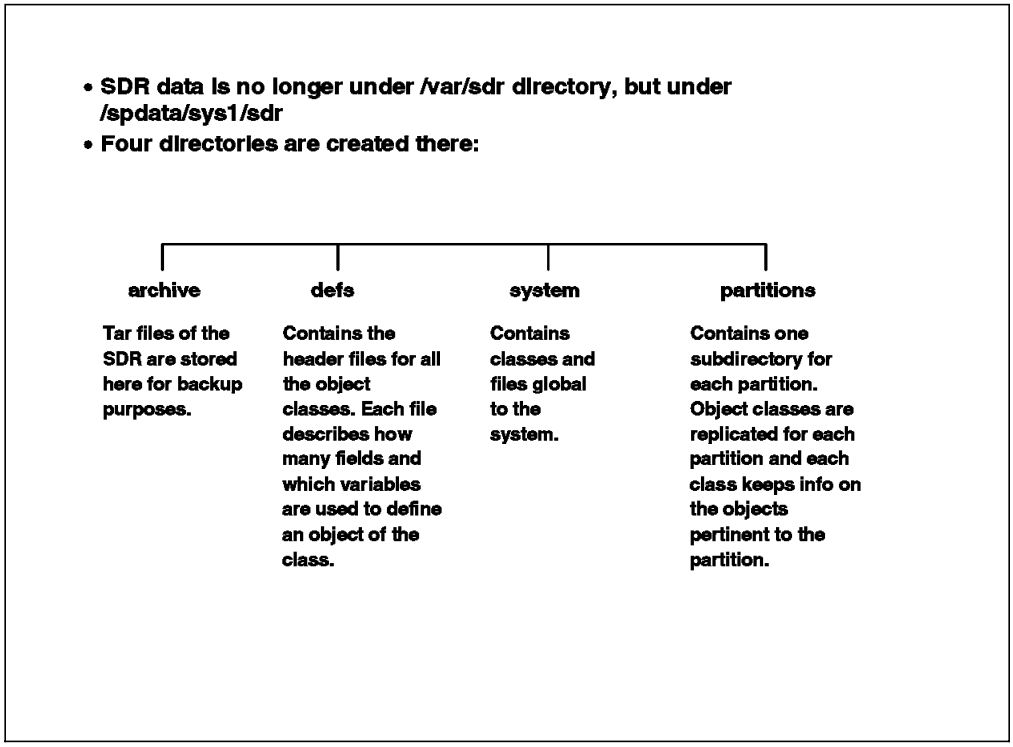


Figure 67. Data Organization of the SDR in PSSP 2.1

The SDR object classes have been split into two types: object classes that are systemwide:

```
/spdata/sys1/sdr/system
```

and object classes that are specific to the partitions:

```
/spdata/sys1/sdr/partitions
```

The templates or definitions for the names of the attributes contained in all the object classes are now found in a separate file from the actual data. These definitions can be found in:

```
/spdata/sys1/sdr/defs
```

Each partition has its own directory identified by its *associated* IP address (IP alias or Control Workstation IP address). In the example given, the following directories exist:

```
/spdata/sys1/sdr/partitions/9.180.6.199
/spdata/sys1/sdr/partitions/9.180.6.198
```

The switch topologies are stored in the respective partition directories in a subdirectory called *files*. In the example given:

```
/spdata/sys1/sdr/partitions/9.180.6.199/files/SDR.expected.top.any.l1.8way1.0isb
/spdata/sys1/sdr/partitions/9.180.6.198/files/SDR.expected.top.any.l1.8way2.0isb
```

The diagram shown in Figure 68 on page 149 gives examples of the object classes that are systemwide and those that are partition-specific.

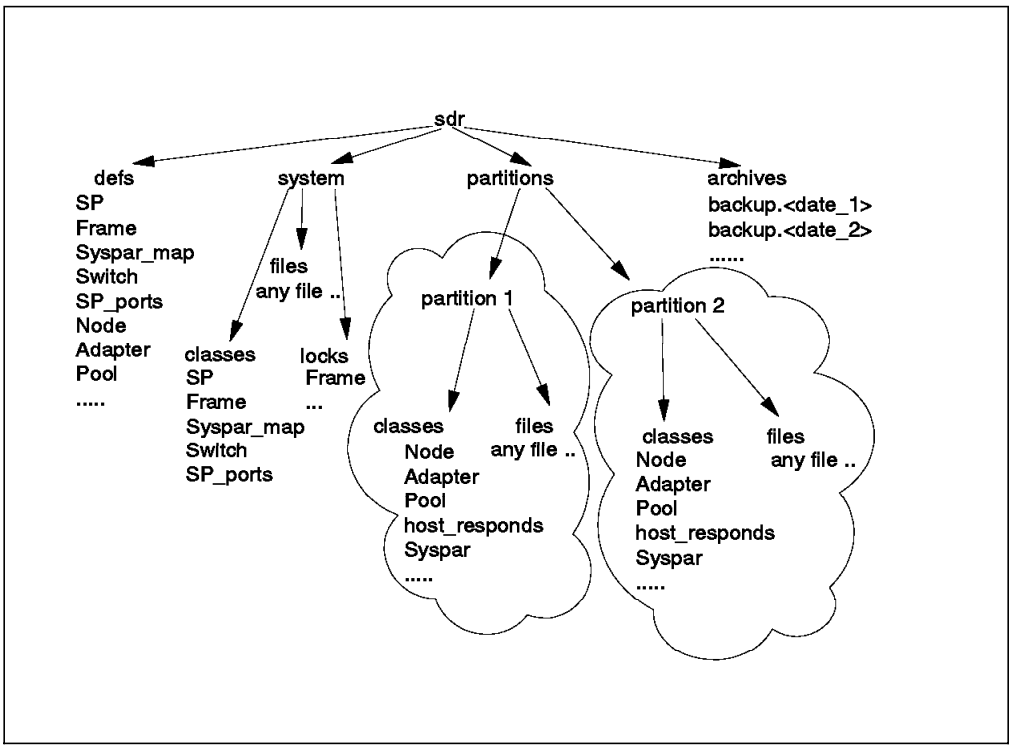


Figure 68. PSSP 2.1 SDR Directory Structure

5.4.3 New Object Classes

Two completely new object classes have been created for System Partitioning. The *Syspar_map* object class has already been mentioned earlier. It is a systemwide or global class that describes which nodes are in which partition. Running the following command:

```
# SDRGetObjects Syspar_map
```

in the example given produces the output:

```
syspar_name  syspar_addr  node_number  switch_node_number  used
spcws        9.180.6.199            1                   0     1
spcws        9.180.6.199            2                   1     0
sppart2      9.180.6.198            3                   2     1
sppart2      9.180.6.198            4                   3     0
spcws        9.180.6.199            5                   4     1
spcws        9.180.6.199            6                   5     1
sppart2      9.180.6.198            7                   6     1
sppart2      9.180.6.198            8                   7     1
spcws        9.180.6.199            9                   8     1
spcws        9.180.6.199           10                   9     1
sppart2      9.180.6.198           11                  10     1
sppart2      9.180.6.198           12                  11     1
spcws        9.180.6.199           13                  12     0
spcws        9.180.6.199           14                  13     0
sppart2      9.180.6.198           15                  14     0
sppart2      9.180.6.198           16                  15     0
```

From this output, it is possible to see that there are two wide nodes and eight thin nodes from the *used* field. The *0* signifies that the slot is not used. Run this command if there are problems with the partitions. If there is any erroneous data, the health of the partitions will be affected. It is possible to edit manually

the object class with a program such as vi, but be sure to copy the object class to a safe place first should the file become inadvertently corrupted.

Also, because the SDR daemon was bypassed during this operation, it is essential to refresh the relevant daemon (that is, the one for that partition) so that the change can take effect. Use the stopsrc and startsrc commands, as shown in previous examples.

The second object class that gets created is *Syspar*, which is a partition-specific class (so there is one for each partition) that contains all the attributes that were entered in the SMIT fields when the partition was customized. Run the following commands if you have problems with a particular partition:

```
# export SP_NAME=<syspar name>
# SDRGetObjects Syspar
```

The data should match the data contained in the custom file that was created during the customization of the partition. Ensure that the data is consistent between the SDR and the custom file, and correct any errors. Following is the data from the SDRGetObjects command for the *default* partition in the example:

```
syspar_name  ip_address  install_image  syspar_dir  code_version
spcws        9.180.6.199 default        /spdata/sys1/sypar_configs/1nsb0isb
/config.8_8/layout.1/syspar.1  PSSP-2.1
```

The default partition is at PSSP 2.1, which is shown by the *code_version* attribute. Here is the same data for the second partition:

```
syspar_name  ip_address  install_image  syspar_dir  code_version
sppart2      9.180.6.198 default        /spdata/sys1/sypar_configs/1nsb0isb
/config.8_8/layout.1/syspar.2  PSSP-2.1
```

This data is consistent with the *custom* file example in the previous examples.

5.4.4 SDR Locking

Now that there is the potential to have multiple SDR daemons accessing the same systemwide or global object classes, there is the requirement to have a persistent locking mechanism to prevent simultaneous updates of object classes. The lock files are created in the following directory:

```
/spdata/sys1/sdr/system/locks
```

The file that is created contains details of the node updating the SDR and the partition to which it belongs.

Whenever the SDR daemons are started up, they check for the presence of any lock files in case they were left behind when an update of the SDR ended abnormally. If any are found at this point, they are removed. This is another good reason why it is a good idea to recycle the SDR daemons when there are apparent SDR problems.

5.4.5 The Restructured SDR

The SDR gets reorganized during the *apply* stage of System Partitioning. Before the new partitions are set up, there are some checks that are carried out by the spapply_config command that carries out the *apply* of the partitioning. This command verifies the following:

- The data in the specified layout directory (see the previous section on *verifying* the configuration for more details).

- The host listed in each of the custom files has the same interface as the *hostname* of the Control Workstation.
- The SDR is up and can answer a call.
- There is exactly one default partition whose name is the same as that stated in the SP object class (that is, host name of the Control Workstation).
- The existing SDR is consistent by calling the spchk_syspars command before setting up the new partitions.
- The command generates a list of affected and unaffected nodes based on the current configuration and the data for the new partitions in the *syspar* directories.
- The VSDs will not span the new partitions by running the verparvsd command.

If any errors are generated from these checks, then the spapply_config command will quit there and report an error. All of the issues have been discussed previously, so run back through the details to rectify the situation. When all these checks have been completed without error, the spapply_config command begins the reorganization of the SDR and daemons as follows:

- Deletes heartbeat and host_responds daemons for partitions that will not exist in the new partitioned environment (as partitioning may previously have been set up on the system).
- Creates the new partitions using the SDRAddSyspar command, which creates the new SDR directories, object classes, and SDR daemons.
- Moves data from the original Node, Adapter, host_responds and switch_responds to the newly-created ones, but only for the affected nodes.
- Opens a custom file in the syspar directory to get the data to put into the *Syspar* object class.
- Updates *Syspar_map* object class with new information for the affected nodes such as the syspar_name and *syspar_addr*.
- After obtaining the code level of the partition, it makes and starts the new heartbeat daemon (as the heartbeat may be in compatibility mode for an AIX Version 3.2.5 partition).
- Makes and starts the new host_responds daemon.
- Adds the path to the layout directory in the *SP* object class.
- Runs the partitionVSDdata command to restructure the VSD data (if it exists).
- Deletes any partitions that no longer exist in the newly-partitioned system by running the SDRRemoveSyspar command, which deletes the SDR daemon and SDR data associated with the partition.
- Sets up the switch in each partition by running Eprimary, Eannotator, and Etopology. This loads the new topology file into the SDR, which is then ready to run the Estart command when the nodes are available.
- Verifies that the data in the SDR is correct by running the command spverify_config, which checks against the original data supplied in the syspar directory.

If there are any errors, then check manually in the relevant SDR object classes. Check that the newly-created data was copied from the syspar directory without any errors, and similarly, that the data got moved from the existing object

classes (such as *Node*, for example). In addition, look at the relevant log files for the SDR which can be found in the following directory:

```
/var/adm/SPlogs/sdr
```

Here there are various types of files. The detailed output from the daemons is found in SDR_config.log. Use the `tail` command to view the last entries made in this file. There are also two files for each SDR daemon that have basic information about the last two invocations of the daemon. These files have the format:

```
sdrlog.<IP address of syspar>.PID
```

5.5 Heartbeat Reorganization

The heartbeat has been mentioned several times already because it is an important part of the administration of an RS/6000 SP and it is also affected by System Partitioning. Now we will look at the heartbeat in more depth.

5.5.1 The Heartbeat before System Partitioning

The heartbeat on an RS/6000 SP is used to monitor connectivity to the nodes and the Control Workstation across the *en0* network interface. It is often regarded as the indicator of whether a node is up or not, but the loss of connectivity across *en0* does not always signify this, since the node may be up but the connectivity on this interface may be lost.

The heartbeat daemon runs on each node and the Control Workstation. At PSSP 2.1, the daemon is called hbd; at PSSP 1.2, it is known as ccst. Run the following commands to check if the daemons are running at PSSP 2.1:

```
# ps -ef | grep hbd
```

and at PSSP 1.2:

```
# ps -ef | grep ccst
```

Here is an example of the output from these commands showing just the process name at PSSP 2.1:

```
/usr/lpp/ssp/bin/hbd -p0 -u
```

and at PSSP 1.2:

```
/usr/lpp/ssp/bin/ccst -p0
```

The ccst daemon will only be running on AIX Version 3.2.5 nodes. These daemons are started from the following inittab entries, at PSSP 2.1:

```
hb:2:once:/usr/bin/startsrc -g hb >/dev/null 2>/dev/console
```

and at PSSP 1.2:

```
hb:2:respawn:/usr/lpp/ssp/bin/hb >/dev/console 2>&1
```

At PSSP 2.1, the heartbeat was brought under SRC control for the same reasons given earlier about the SDR daemons.

Each daemon *heartbeats* or *pings* to its nearest neighbor in a ring fashion from the highest to the lowest IP address. The Control Workstation is known as the *group leader* because it coordinates the activity of all the other heartbeat daemons on the nodes. You can logically view the Control Workstation as starting the ping to the node with the highest IP address, which then *pings* to the

node with the next highest IP address (and so on), until the ring is completed by the node with the lowest IP address *pinging* to the Control Workstation.

If there is no response to several retries of the *ping* packet within the timeout period, that heartbeat daemon will notify the *group leader* that connectivity has been lost over *en0*, which will then take that node out of the ring.

The *group leader* will notify all the nodes of this action so that the ring can be re-established by missing out the node from which there is no response over *en0*. The heartbeat daemon on the Control Workstation will pass the information about this node up to the host_responds daemon on the Control Workstation, which in turn passes it up to the SDR daemon to update the *host_responds* object class of the SDR.

The spmon command uses this object to find out the *host_responds* status of all the nodes. It may take a few seconds before the loss of connectivity is reflected by the node changing from green to red on the spmon GUI due to the timeout period mentioned before and the time taken for spmon to again read the object class and refresh the *host_responds* GUI.

The heartbeat daemon on the Control Workstation will send out regular *ping* packets (known as *proclaim* packets) to any nodes that have been marked as missing from the ring in the *host_responds* object class, to test whether connectivity has returned, and will reintegrate them back into the ring if that is the case.

If a node goes *red* on the spmon GUI, run the following command:

```
# SDRGetObjects host_responds
```

Here is an example of the output:

```
node_number   host_responds
          1              1
          3              1
          5              1
          6              1
          7              1
          8              1
          9              1
         10              1
         11              1
         12              0
```

In this example, all the nodes have *en0* connectivity except node 12, which has an entry of *0* (signifying a loss of connectivity).

In such a case, first test the connectivity manually by *pinging* the *en0* interface. If the *ping* is successful, wait a few seconds to see if spmon will refresh. If it does not, then check that the heartbeat, SDR, and heartbeat daemons are running on the Control Workstation, and that the heartbeat daemon is running on the problematic node.

Examples have already been given for heartbeat and SDR, but following is an example for the host_responds daemon. It shows how to find out if it is running, the inittab entries, and how to start it if it has died:

```
# ps -ef | grep hr
```

Following is an example of the process name output from this command:

```
/usr/lpp/ssp/bin/hrd
```

The inittab entry:

```
hr:2:once:/usr/bin/startsrc -g hr >/dev/null 2>/dev/console
```

If it has died or did not start on boot, run:

```
# startsrc -g hr
```

The host_responds daemon (like the SDR daemon) only runs on the Control Workstation. Remember that if the problematic node is running AIX Version 3.2.5, then check for the *ccst* process instead of the *hbd* process.

If all the daemons are running, it is still worth recycling them all just in case any of the daemons are no longer functioning as they should.

If there is no connectivity across *en0*, then test the connectivity across the other network interfaces to the problem node. Test to see if a console can be opened across the serial link. If there is connectivity across other interfaces, and the node is up and working, then resolve the network issues with the *en0* interface. Here are some possible scenarios:

- An application or process is generating a large amount of traffic across *en0* so that the heartbeat times out
  - The internal Ethernet should ideally be dedicated to *SP*-related traffic for this reason.
- The interface has gone down or been corrupted
  - Check with the `ifconfig` command and remove and rebuild the interface with the `rmdev` and `mkdev` commands if necessary.

If there is no connectivity across any of the interfaces, then check to see what the status is of any users that are already logged in. If their sessions are *hung* (that is, if there is no response to typing anything on the keyboard), then check the *LED* display for the node using the `spmon` GUI.

If the node has crashed, then there will be either an *888* LED displayed, or an associated crash code (such as *0c9*, for example). In this case, reboot the node after ensuring that it has finished taking a system dump and contact IBM support for assistance in analyzing the dump data.

If the node shows no sign of having crashed but there is absolutely no response at all to any commands, then force a system dump (according to the instructions given in Chapter 8, “Producing a System Dump” on page 201), reboot the node, and contact IBM support.

If the node responds to commands but does so in a fashion that is slower than normal, then it may be possible to analyze the performance at this time to identify what might be causing this problem.

5.5.2 The Heartbeat after System Partitioning

During the *apply* of the partitions, multiple heartbeat and host_responds daemons get created and started. The `spapply_config` command uses the information in the *Syspar_map* object class to identify which subsystems to build. Each partition has one heartbeat and one host_responds daemon. Unlike the SDR daemons, it is not possible to identify which daemon services which

partition by looking at the process table. However, you can run the following commands to find out this information:

```
# lssrc -g hb
# lssrc -g hr
```

Following is an example of this output for the heartbeat:

```
Subsystem         Group          PID      Status
 hb.sppart2       hb             14302    active
 hb.spcws         hb             15332    active
```

and for the host_responds:

```
Subsystem         Group          PID      Status
 hr.sppart2       hr             16582    active
 hr.spcws         hr             16846    active
```

The partition name is appended to the daemon name to give a unique subsystem name from which to determine which daemon is servicing which partition.

With these distinct subsystems on the Control Workstation, each partition has its own separate and distinct *heartbeat* ring. These rings function completely independently from each other with no *heartbeat* traffic flowing between the rings.

Figure 69 shows this structure.

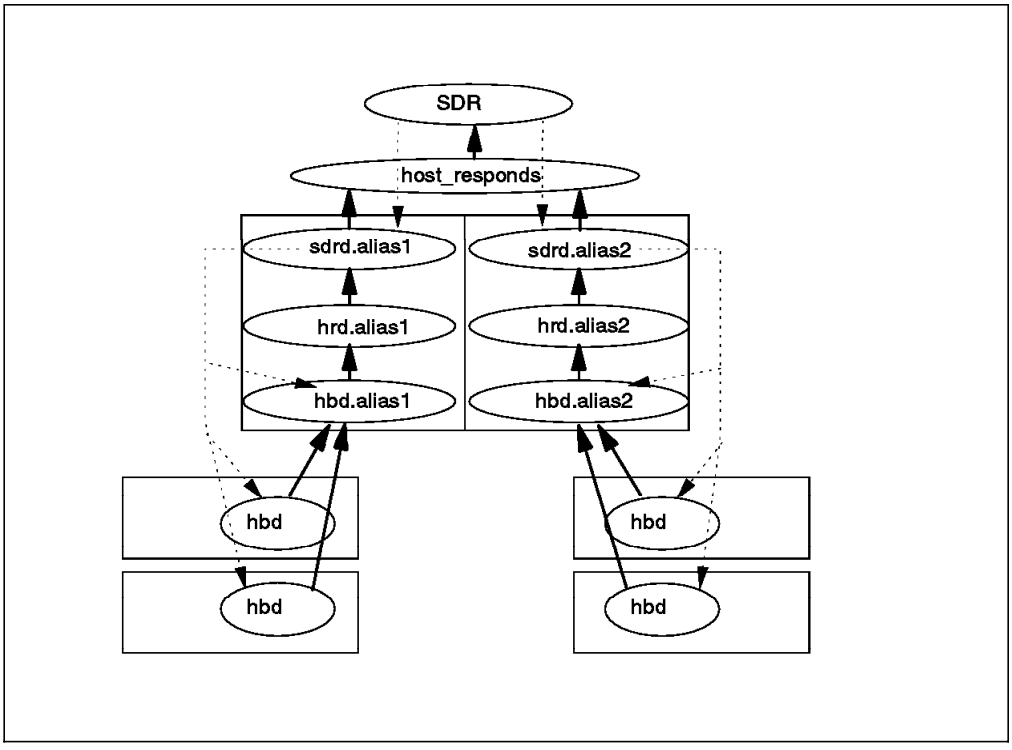


Figure 69. Example of the Heartbeat Subsystem after System Partitioning

If there are any AIX Version 3.2.5 partitions, the heartbeat daemons on the Control Workstation that service these will be started up in *compatibility mode* (but they are still hbd daemons) so that they can communicate with the ccst daemons on the nodes.

If there are problems with the heartbeat daemons after System Partitioning, use the same techniques that have already been described in this chapter.

If there is a problem in one partition, it is possible to work just with the relevant subsystem, rather than with all of them. In the example, the following commands can be run to stop and start the heartbeat daemon:

```
# stopsrc -s hb.spcws
# startsrc -s hb.spcws
```

5.5.3 Daemons and Scripts

As with the SDR daemons, the SRC ensures that multiple instances of the daemons are not running at the same time. It uses the following scripts to start the respective daemons:

```
/usr/lpp/ssp/bin/hb
/usr/lpp/ssp/bin/hr
```

These scripts can be used manually to control the daemons and there are many options that can be used. When using these, be careful not to start multiple daemons because this will result in unpredictable *host_responds information*.

In the example, the heartbeat daemon can be put into debug mode on the *spcws* partition by issuing the following commands:

```
# export SP_NAME=spcws
# /usr/lpp/ssp/bin/hb debug
```

This heartbeat daemon will now send all stdout to the console to make debug work easier when experiencing problems with this subsystem. Since stderr always goes to the console by default, this may provide additional information that assists in resolving heartbeat problems. To turn this option off, issue:

```
# /usr/lpp/ssp/bin/hb debug off
```

Rather than sending this output to the console, it is possible to send this output to a file by changing an attribute in the ODM. The SRC references an object class in the ODM called *SRCsubsys* which contains information about all the subsystems under its control. The following command can be run for any partition:

```
# odmget -q "subsysname=hb.<syspar_name>" SRCsubsys
```

For the example of the partition spcws, the output is:

```
SRCsubsys:
        subsysname = "hb.spcws"
        synonym = ""
        cmdargs = "-spname spcws -splevel PSSP-2.1 "
        path = "/usr/lpp/ssp/bin/hb"
        uid = 0
        auditid = 0
        standin = "/dev/null"
        standout = "/dev/null"
        standerr = "/dev/console"
        action = 2
        multi = 0
        contact = 3
        svrkey = 0
        svrmtype = 0
        priority = 1
        signorm = 0
```

```
        sigforce = 0
        display = 1
        waittime = 20
        grpname = "hb"
```

Turning on debug merely changes the ODM attribute for stdout and refreshes the relevant daemon. Follow the next example, substituting the partition name in order to redirect output to a file:

```
# odmget -q "subsysname=hb.spcws" SRCsubsys > <filename>
# cp /etc/objrepos/SRCsubsys /etc/objrepos/SRCsubsys.orig
# odmdelete -q "subsysname=hb.spcws" -o SRCsubsys
```

Edit <filename>, and change the stdout and stderr to a file where the data is to go, then:

```
# odmadd <filename>
# stopsrc -s hb.spcws
# startsrc -s hb.spcws
```

There are no specific heartbeat log files, so this method may come in useful if you are experiencing problems that cannot readily be resolved. It is worth looking at the file:

```
/var/adm/SPlogs/SPdaemon.log
```

since there may be a reference to the *heartbeat* here.

The host_responds daemon does have a specific log, which is:

```
/var/adm/SPlogs/spmon/hr.log
```

The heartbeat daemons can provide information to the VSD daemons `had` and `hcd` if they are installed. The heartbeat daemons will start up with the `-p1` flag set if they are required to *wait* for these VSD daemons. Otherwise, the heartbeat daemons get started with the `-p0` flag and this operation is bypassed. The examples given earlier in the chapter show how this can be viewed in the process table.

5.6 Resource Manager Reorganization

It was mentioned earlier that Resource Manager gets partitioned also. In fact Resource Manager views each system partition as a separate logical partition. Each partition that runs a Resource Manager has a completely separate configuration file called:

```
jmd_config.<syspar_name>
```

Each partition has its own primary and backup Resource Manager. As a result, nodes can only be allocated to resource pools within their own partition.

The `jm` commands all run against the partition specified by the SP_NAME variable. This variable can be set to whichever partition the commands need to be run against.

There is no difference in the methods used to resolve Resource Manager problems on a partitioned system except that care needs to be taken so that the SP_NAME is set to the partition you are experiencing problems with.

Chapter 6. Error Logging

The RS/6000 SP uses the AIX and BSD error logging mechanism to handle error generation and error reporting. The trend is to use AIX error logging exclusively, but PSSP 2.1 uses some public domain codes, such as AMD, NTP, Supper, and so on. Therefore the BSD syslog daemon must be present.

The PSSP code also provides a specific daemon for SP-specific error log entries like power supplies, fans, and (in general) errors detected by the hardmon daemon. This splogd daemon also interacts with the BSD syslog daemon and with the AIX errdemon daemon.

6.1 Error Logging Overview

The error logging facility records hardware and software failures in the error log for information purposes or for fault detection and correction. In AIX Version 4.1, some of the error log commands are delivered in an optionally-installable package called bos.sysmgt.serv_aid.

The base system bos.rte includes the services for logging errors to the error log file and the errlog subroutines, the errsave kernel service, the error device driver /dev/error, the error daemon, and the errorstop command.

The commands required for licensed program initialization, errinstall, and errupdate are also included in bos.rte.

System dump commands are also included in the bos.sysmgt.serv_aid (for example, the sysdumpstart command).

To determine if this package is installed, run the command:

```
# lslpp -l | grep bos.sysmgt.serv_aid
```

If the package is installed, a line containing the following will be displayed:

```
bos.sysmgt.serv_aid    4.1.4.0    COMMITTED    Software Error Logging and
```

If the package is not installed, you can use SMIT or use the installp command:

```
installp -xad <installation device> bos.sysmgt.serv_aid
```

Because the errclear, errdead, errlogger, errmsg, and errpt commands are part of the optionally installable Software Service Aids package (bos.sysmgt.serv_aid), you need the Software Service Aids package to generate reports from the error log or delete entries from it.

The error logging process begins when an operating system module detects an error. The error-detecting segment of code sends error information to the errsave kernel or the errlog subroutine, where the information is written to the /dev/error special file. Here, a timestamp is added to the collected data.

The errdemon daemon constantly checks /dev/error for new entries, and when new data is written, the daemon will perform a series of operations. Before an entry is written to the error log, the errdemon compares the label sent by the kernel or application code to the contents of the Error Record Template Repository. If the label matches an item in the repository, the daemon collects additional data from other parts of the system.

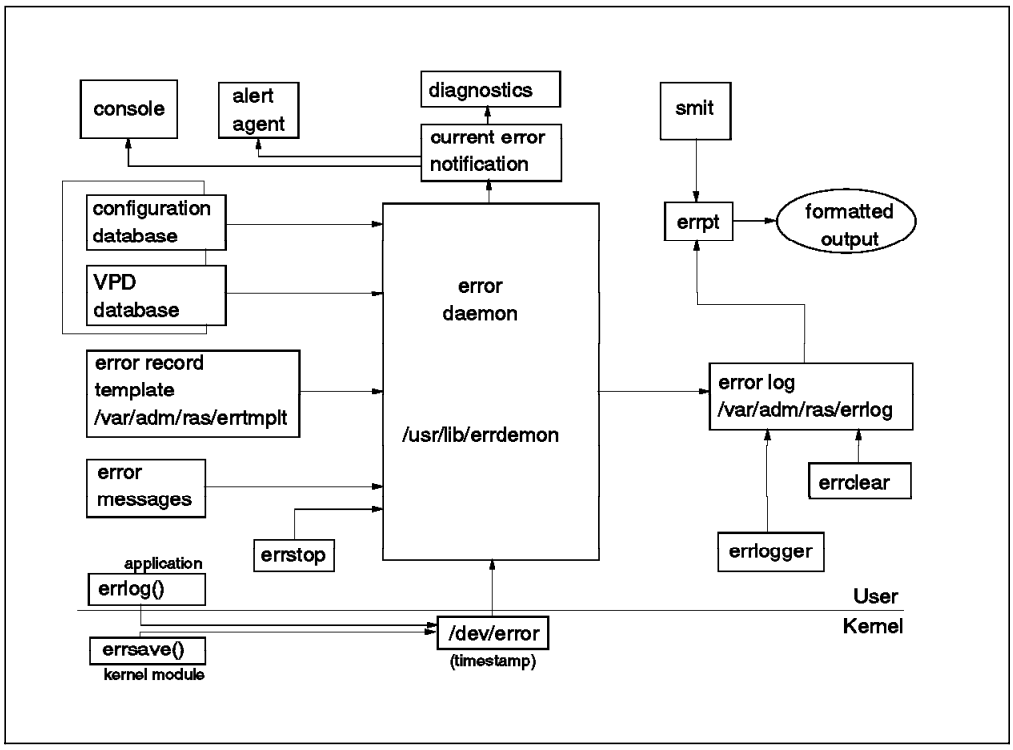


Figure 70. Error Logging Components

To create an entry in the error log, the errdemon retrieves the appropriate template from the repository, the resource name of the unit that caused the error, and detailed data. If the error signifies a hardware error and hardware vital product data (VPD) exists, the daemon retrieves the VPD from the ODM.

When accessing the error log through `smit` or with the command `errpt`, the error log is formatted according to the error template repository and represented in either a summary or detailed report.

The `diag` command uses the error log in part to diagnose hardware problems. To correctly diagnose new system problems, the system deletes hardware-related entries older than 90 days from the error log and software-related entries 30 days after logging. See the `errclear` command lines in the crontab file.

6.1.1 Terms Used by the Error Logging Facility

Error ID — A 32-bit CRC hexadecimal code used to identify a particular failure. Each error record template has a unique error ID.

Error label — The mnemonic name of an error ID.

Error log — The file that stores instances of errors and failures encountered by the system.

Error log entry — A record in the system error log that describes a hardware or software failure or an operator message. An error log entry contains captured failure data.

error log template — A description of what will be displayed when the error log is formatted for a report; includes information on the type and class of the error, probable causes, and

recommended actions. Collectively, the templates comprise the Error Record Template Repository.

6.1.2 Error Logging Commands

errclear Deletes entries from the error log. This command can erase the entire error log. Removes entries with specified error ID numbers, classes, or types.

errdead Extracts errors in the /dev/error buffer captured in the system dump. The system dump will contain error records if errdemon was not active prior to the dump.

errdemon Reads error records from /dev/error and writes error log entries to the system error log. The errdemon also performs error notification as specified in the notification objects in the ODM. The daemon is started automatically during system initialization.

errinstall Can be used to add or replace messages in the error message catalog. Provided for use by software installation procedures. The system creates a backup file called <file>.undo. The .undo file allows you to cancel the changes you made by issuing the errinstall command.

errlogger Writes an operator message entry to the error log.

errmsg Implements error logging in applications. The errmsg command will list, add or delete messages stored in the error message catalog. With this command you can add text to the Error Description, Probable Cause, User Cause, Failure Cause, Recommended Action, and Detailed Data message sets.

errpt Generates an error report from entries in the system error log. The report can be formatted to give a single line for each entry, or as a detailed listing.

errstop Stops errdemon, which is initiated during system initialization. Running the errstop command also disables some diagnostic and recovery functions of the system.

errupdate Adds or deletes templates in the Error Record Template Repository. Modifies the Alert log and Report attributes of an error template. Provided for use by software installation procedures.

6.1.3 Error Log Files

/dev/error Provides standard device driver interfaces required by the error log component.

/dev/errorctl Provides non-standard device-driver interfaces for controlling the error logging system.

/usr/include/sys/err_rec.h Contains structures defined as arguments to the errsave kernel services and the errlog subroutine.

/var/adm/ras/errlog	Stores instances of errors and failures encountered by the system.
/var/adm/ras/errtmplt	Contains the Error Record Template Repository.

The /dev/error and /dev/errorctl special files support the logging of error events. Minor device 0 (zero) of the error special file is the interface between processes that log error events and the errdemon daemon. Error records are written to /dev/error by the errlog library routine and the errsave kernel service. The error special file timestamps each error record entry.

errdemon opens /dev/error for reading. Each read retrieves an entire error record. The format of the error records are described in the header file /usr/include/sys/err_rec.h.

Each time a error is logged, the error ID, the resource name, and the timestamps are recorded in Non Volatile Random Access Memory (NVRAM) so that in the event of a system crash, the last logged error is not lost. When the error file is started, the last error entry is retrieved from NVRAM.

The standard device driver interfaces (open, close, read, and write) are provided for the error file. The error file has no *ioctl* functions. The ioctl function interface for the error special file is provided by the errorctl special file. This interface supports stopping the error logging system, synchronizing the error logging system, and querying the status of the error special file.

6.2 SP Error Logging

The RS/6000 SP uses both the AIX Error Logging facilities and the BSD syslog, as well as a number of function-specific log files to record error events on each node.

Commands and SMIT panels are available to perform general log management. For managing Syslog and AIX Error Log, it is necessary to install the ssp.sysman option.

Error logging is the writing of information to persistent storage to be used for debugging purposes. This type of logging is for subsystems that perform a service or function on behalf of an end user. The subsystem does not communicate directly with the end user and therefore needs to log events to a storage location. The events that are logged are primarily error events.

Error logging for the SP uses BSD syslog and AIX Error Log facilities to report events on a node basis. The System Monitor, the High Performance Switch, SP Switch, and the Resource Manager components use this form of error logging.

Error log entries include a DETECTING MODULE string that identifies the software component, module name, module level, and line of code or function that detected the event that was logged. The information is formatted depending on the logging facility the user is viewing. For example the AIX Error Log facility information appears as follows:

```
DETECTING MODULE
LPP=<LPP name>  Fn=<filename>  <ID_level_of_the_file>  L#=<line number>
```

The BSD syslog facility information appears as follows:

```
<timestamp, hostname, ID, PID>
LPP=<LPP name>  Fn=<filename>  <SID_level_of_the_file>  L#=<line number>
```

Effect of Not Having a Battery on Error Logging

In a regular RS/6000 system, a battery is installed to maintain NVRAM. On an RS/6000 SP system, there is no battery and NVRAM may be lost when the node is powered off. AIX writes the last error log entry to NVRAM. During system startup, the last entry is read from NVRAM and placed in the error log when the error daemon is started. This last error log entry may be important in diagnosis of a system failure.

On SP wide nodes, the NVRAM does have power to it as long as the node is plugged into the frame and the frame is plugged into a power source with power.

On SP thin nodes, NVRAM is lost whenever the node is powered down. If the last error entry is desired, the thin nodes should not be powered off. They should reboot with the key in “Normal” position.

6.2.1 Install and Configure Error Log

Log management functions are built upon the sysctl facility, which uses the SP authentication services. Generating parallel AIX Error Log and BSD syslog reports and performing general log viewing require that the user issue the `kinit` command to be identified to the SP authentication services. All other log management commands additionally require that the user be defined as a principal in the /etc/logmgt.acl file. All users defined in this file must also be placed in the authentication (PSSP or AFS) database as a principal.

Note: Log management mostly consists of administrative tasks normally requiring root authority, and require a user defined in the logmgt.acl file to execute commands as the root user.

The following is an example of the /etc/logmgt.acl file:

/etc/logmgt.acl

```
#acl#
# This sample acl file for log management commands contains
# a commented line for a principal
#_PRINCIPAL root.admin@HPSSL.KGN.IBM.COM
# The following principal is added to be able to trim SPdaemon.log
# from cleanup.logs.ws
_PRINCIPAL rcmd.localhost
_PRINCIPAL root.admin@HPSSL.KGN.IBM.COM
_PRINCIPAL joe@HPSSL.KGN.IBM.COM
```

The file contains entries for the user root as principal *root.admin* and user Joe as principal *joe*, giving them authority to execute log management commands.

The log management server functions executed by sysctl are located in /usr/lpp/ssp/sysctl/bin/logmgt.cmds. During system initialization, an include statement for this file is added to the default systctl configuration file /etc/sysctl.conf.

If you want to use an alternate sysctl configuration file, it must be updated with a statement to include the logmgt.cmds file, and the sysctld daemon must be restarted to activate the change.

The following is an example of the /etc/sysctl.conf file:

/etc/sysctl.conf

```
#
# (C) COPYRIGHT IBM CORP.  1993
#
# ALL RIGHTS RESERVED
#
# US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION
# OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH
# IBM CORP.
#

# This points to where sysctl resides
create var buildTop /usr/lpp/ssp AUTH

# Set up the environment for the rcmd sample programs
set env(PATH) /bin:/usr/bin:/usr/ucb:/etc:/usr/etc:/usr/sbin

# Include the help commands, but override the default helpPath
create class help $buildTop/help/help.cmds
help:setPath $buildTop/help

# Include system information commands
create class sys $buildTop/samples/sysctl/sys.cmds

# Include process manipulation commands
include $buildTop/samples/sysctl/process.cmds

# Include file system commands
include $buildTop/samples/sysctl/filesys.cmds

# Include more file system commands (pdf and pfck)
include $buildTop/sysctl/bin/pdfpfck.cmds

# Include pfps commands
include $buildTop/sysctl/bin/pfps.cmds

# Include rcmds
create class rcmds $buildTop/samples/sysctl/rcmds/rcmds.cmds

# Include LogMgt commands
include /usr/lpp/ssp/sysctl/bin/logmgt.cmds
```

6.2.2 AIX Error Log Facility

SMIT menus as well as commands are provided to help you manage the AIX Error Log facility.

When you execute commands related to AIX Error Log Management, you can specify whether the command should be executed on all nodes in the current system partition, or you can specify node names or the name of a file containing a list of node names. The default is the local node.

To access the AIX Error Log SMIT menu, enter:

```
smit sperrlog
```

Trim AIX Error Logs

You can trim records from error logs on a set of nodes. The fast path invocation that cleans the Error Log menu is:

```
smit perrclear
```

Configuring the AIX Error Log

You can display the configuration parameters of the AIX Error Log to the local node. The fast path invocation for the Show Characteristics of the Error Log menu is:

```
smit perrdemon_shw
```

You can alter one or more of the configuration parameters for the AIX Error Log on a set of nodes. Because the additional entries are generated by SP System software, the AIX Error Log file size should be a minimum of 4 MB.

The fast path invocation for the Change Characteristics of the Error Log menu is:

```
smit perrdemon_chg
```

Managing Error Templates

You can create a new error template in the Error Template repository for logging errors. The fast path invocation for the Add an Error Template menu is:

```
smit padd_et
```

You can remove a template from the Error Template Repository using the fast path invocation for the Error Template menu:

```
smit prem_et
```

You can display entries from the Error Template Repository to the local host by using the fast path invocation for the Show an Error Template menu:

```
smit pshw_rt
```

6.2.3 BSD syslog Facility

SP error logging utilizes the BSD syslog facility for recording error events. Some applications use syslog for logging errors and other events. Administrators also find it desirable to be able to list error log messages and syslog messages in one single report.

To access the syslog menu in SMIT, enter:

```
smit spsyslog
```

Generating Reports on BSD syslog Log Files

The syslogd daemon, which logs the errors, is configured with a file designating filters for the incoming messages to determine their destination. The default configuration file is /etc/syslog.conf. BSD syslog errors are classified by the facility that is issuing the error and by the error′s priority value. Refer to syslogd daemon later in this topic for a description and overview of facility and priority values. Entries in the configuration file determine the destination for each error message based on these values. Destinations can be a file, user ID or the syslogd daemon on another machine. We recommend that error messages be kept locally rather than forwarding to a remote syslogd daemon on another machine, because of the increased network traffic. You can use file collections to maintain consistent configuration of the syslog facility.

A syslog file has the following format:

```
MMM  DD  HH:MM:SS  node_name  resource{pid}:  mesg
```

Where:

MMM DD HH:MM:SS Is the timestamp: Month Day Hour:Minute:Second

node_name The name of the node that the error occurred on

resource The name of the failing system

pid	The logged process ID of the failing resource (optional)
msg	A free form error message

Important

Note that syslogd does not log the year in a record's timestamp. The comparisons for start and end times are done on a per record basis and can cause unexpected results if the log is allowed to span more than one year.

Errors logged by SP components will contain in the message section the following additional information pertaining to the logging resource:

LPP	LPP name
Fn	File name
SID	SID level of the file
L#	Line number or function

Note: You must be Kerberos-authenticated to run the `psyslrpt` command.

The fast path to invoke for the Generate a Syslog Report (SMIT) menu is:

```
smit spsyslrpt
```

To report to the local node all records logged by ftp for all syslog files on nodes in the current system partition, enter:

```
psyslrpt -a -r ftp
```

To report all records that logged to files selected for the daemon and user facilities starting on August 9 and to report the records to the local node from the nodes sp21n1 and sp21n2, enter the following command at the command line:

```
psyslrpt -w sp21n1,sp21n2 -f daemon,user -s08090000
```

The following is an example of the /etc/syslog.conf file:

```
# @(#)341.9  src/bos/etc/syslog/syslog.conf, cmdnet, bos411, \
9428A410j 6/13/93 14:52:39
#
# COMPONENT_NAME: (CMDNET) Network commands.
#
# FUNCTIONS:
#
# ORIGINS: 27
#
# (C) COPYRIGHT International Business Machines Corp.  1988, 1989
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# /etc/syslog.conf - control output of syslogd
#
#
# Each line must consist of two parts:
#
# 1) A selector to determine the message priorities to which the
#    line applies
# 2) An action.
#
# The two fields must be separated by one or more tabs or spaces.
#
# format:
#
# <msg_src_list><destination>
#
# where <msg_src_list> is a semicolon separated list of <facility>. \
  <priority>
# where:
#
# <facility> is:
# * - all (except mark)
# mark - time marks
# kern,user,mail,daemon,auth, ... \
  (see syslogd (AIX Commands Reference))
#
# <priority> is one of (from high to low):
# emerg/panic,alert,crit,err(or),warn(ing),notice,info,debug
# (meaning all messages of this priority or higher)
#
# <destination> is:
# /filename - log to this file
# username,username2,...  - write to user(s)
# @hostname - send to syslogd on this machine
# * - send to all logged in users
#
# example:
# "mail messages, at debug or higher, go to Log file.  \
   File must exist."
# "all facilities, at debug and higher, go to console"
# "all facilities, at crit or higher, go to all users"
#  mail.debug            /usr/spool/mqueue/syslog
#  *.debug               /dev/console
#  *.crit                *
daemon.notice            /var/adm/SPlogs/SPdaemon.log
```

Figure 71. /etc/syslog.conf file

The following is an example of the /etc/syslog.pid file:

```
3216
```

Figure 72. /etc/syslog.pid file

The psyslrpt Command

You can use the psyslprt command to generate reports of log entries in the log files generated on a set of nodes by syslogd. Options allow you to select the files and records that will be reported.

Options and Parameters

-a	Generates the report on all nodes in the system partition.
-e	Reports on records before endtime (MMddhhmm).
-f facilities	Uses the facilities list to parse the syslog.conf file.
-g config	Specifies the use of an alternate syslog.conf file.
h	Displays usage information.
-l logs	(Lowercase L) Reports on the list of log files, if the syslog.conf file is not parsed.
-n nodes	Reports records matching the nodes.
-p priority	Uses priority value to pass the syslog.conf file.
-s starttime	Reports records created after starttime (MMddhhmm).
-w hosts	Runs the command on the file or list of host names. If the argument begins with a slash (/), it is interpreted as a file containing a list of nodes to execute the command on. Otherwise it can be a list of host names.

Note: If neither the -a nor -w options are used, psyslprt defaults to the local node.

6.2.4 Trimming syslog Files

The size of the syslog log files is *not* configurable and will continue to grow until you trim it manually.

The syslogd daemon is stopped during the trimming process and then restarted with either the default configuration file (/etc/syslog.conf) or an alternate file.

To run the psyslclr command, you must be defined as a Kerberos principal in the /etc/logmgt.acl file.

The fastpath to invoke the Trim Log Files SMIT menu is:

```
smit psyslclr
```

To trim all records older than 30 days from the log file /var/adm/msgs on the local node, you can perform this command:

```
psyslclr -y 30 -l /var/adm/msgs
```

To trim all records from all log files found in the alternate syslog configuration file /var/adm/syslog.conf, perform this command:

```
psyslclr -g /var/adm/syslog.conf -y 0
```

You can add the psyslclr command to the crontab to perform a scheduled syslog trimming.

On the Control Workstation, psyslclr is used to trim daemon facility messages older than 6 days. This is done in /usr/lpp/ssp/bin/cleanup.logs.ws, which is run from the Control Workstation's crontab file.

The SP system uses the crontabs file to periodically update file collections and clean up log files. The installation process appends to the crontabs files in /var/spool/cron/crontabs/root on the Control Workstation and each of the nodes. The files contain different entries depending on their locations.

Both files on the Control Workstation and the nodes contain an entry to clean up the logs. The following example shows the crontab entries for root on a Control Workstation:

```
0 11 * * * /usr/bin/errclear -d S,O 30
0 12 * * * /usr/bin/errclear -d H 90
01 5 * * * /usr/lpp/diagnostics/bin/run_ssa_ela 1>/dev/null 2>/dev/null
0 * * * * /usr/lpp/diagnostics/bin/run_ssa_healthcheck 1>/dev/null 2>/dev/null
0 0  * * *  /usr/lpp/ssp/bin/cleanup.logs.ws
```

If you want the same crontab file on all SP nodes, you must update all the crontab files across the SP system. One way to do this is the following:

1. Get a crontab file from one of the node:

   ```
   rsh hostname crontab -l > /tmp/crontab.nodes
   ```

2. Edit /tmp/crontab.nodes

3. Propagate the changed crontab file to all SP nodes:

   ```
   dsh -a rcp root@mynode:/tmp/crontab.nodes /tmp/crontab.nodes
   dsh -a "crontab < /tmp/crontab.nodes"
   ```

The psyslclr Command

You can use the psyslclr command to delete log entries in the log files generated by syslogd. There are options that allow you to select the files and records that are trimmed, as follows:

Options and Parameters

-a	Trims logs on all nodes in the system partition.
-e	Trims records before endtime (MMddhhmm).
-f facilities	Uses the facilities list to parse the syslog.conf file.
-g config	Specifies the use of an alternate syslog.conf file.
h	Displays usage information.
-l logs	(Lowercase L) Reports on the list of log files, if the syslog.conf file is not parsed.
-n nodes	Reports records matching the nodes.
-p priority	Uses priority value to pass the syslog.conf file.
-s starttime	Trims records created after starttime (MMddhhmm).
-w hosts	Runs the command on the file or list of host names. If the argument begins with a slash (/), it is interpreted as a file containing a list of nodes to execute the command on. Otherwise it can be a list of host names.

The -f and -p options can be used to control selecting files in the configuration file. All files found in the configuration file will be trimmed if both options are not used. If neither the -a nor -w options are used, psyslclr defaults to the local node.

6.2.5 Maintain Error Logs

View error logs

```
smit splogview
```

The contents of each file entry and output of each command entry on the target nodes in the log table amd.tab can be displayed to the stdout of the node from which you are executing the splm command. To do this, issue the following from the command line:

```
splm -a view -t /spdata/sys1/logtables/amd.tab
```

Archive error logs

You can create, remove, or gather error log archives to a central location where they can be (optionally) written to tape or mailed to another location.

To access the archive Logs SMIT menu, enter:

```
smit sparchive
```

The fast path to invoke to the Create Archive menu is:

```
smit spcreate_archive
```

To create an archive on each target node in the directory /var/archives/arc_weekly.tab, issue from the command line:

```
splm -a archive -t /spdata/sys1/logtables/weekly.tab -c -d /var/archives
```

The fast path to invoke to the Remove Archives menu is:

```
smit spremove_archive
```

To remove all files and directories in and including /var/archives/arc_weekly.tab, issue from the command line:

```
splm -a archive -t /spdata/sys1/logtables/weekly.tab -r -d /var/archives
```

The fast path to invoke to the Gather Archives menu is:

```
smit spgather_archive
```

To gather the compressed tar files created on each target node in the directory /var/archives/arc_weekly.tab to the directory /var/logrepos on the local node, issue from the command line:

```
splm -a gather -k archive -t /spdata/sys1/logtables/weekly.tab \
-d /var/archives -l /var/logrepos
```

Note: The archive and gather functions of `splm` require that you have a Kerberos principal defined in the /etc/logmgt.acl file. The command then runs as root on all target nodes.

The `view` function requires that you are Kerberos-authenticated to a valid user ID on the nodes that you are executing on. The server switches IDs from root to your authenticated ID before executing.

The splm Command

With this command, you can execute a number of log management functions on a single host or a set of hosts in parallel. The splm functions are driven by a log table file that contains the target node designations and associated files and the commands to execute. The table format is as follows:

```
#comment line
<target nodes>:  <file or command>
```

The target node portion of a table stanza is a comma-delimited list with no blanks. The values in the list are interpreted as:

1. If the value begins with a slash (/), it is a file containing a list of node names, one per line.
2. If the value is an exclamation point (!), it refers to the local host.
3. Any string not matching list items 1 or 2 is interpreted as a node name.

Options and Arguments

-a action Specifies the function to perform: archive, check, gather, service, or view.

-c Creates a compressed tar file. The tar file will be named node.tar.Z.

-d dir Is the path where the archive collection will be stored on each node. The default is /var/adm/archives.

-k type For the gather function only, this option indicates whether a service collection or archive is being collected.

-l cfs Specifies the path on the local node where the archive collection should be gathered.

-r Removes the archive on each node.

-s Staggers collection to a mail or to a device.

-t Specifies the input table of nodes and commands.

Functions

Archive

The archive function copies files and redirects command output as specified in the input table to the top level directory on each node. The -c option creates a compressed tar file of the data named /topdirectory/node_name.tar.Z. The -r option removes an archive by removing all files down from the top level directory.

Gather

The gather function moves archive tar files to a central location on the executing node. The -r option removes the archive collection on each remote node only after the tar file was copied successfully to the central location. If the node.tar.Z file is not found, the gather function will try to create one. Gathered tar files can be copied to tape or disk or mailed.

Service

The service function first calls the snap command to gather system data to the top level directory if the -p option is used. The snap command creates a set of subdirectories based on the -p arguments. The additional data defined in the table data is then collected in the other subdirectory created by the snap command. If the -p option is not used, the data will be collected in the other subdirectory. Note that you must have root authority to execute the snap command.

View

The view function displays the output of the command or contents of the entries in the input table to the local host. The following is an example of the /spdata/sys1/amd.tab file:

```
#
# This is a sample service collection table for amd problems.
# To use:
#   1.  Uncomment lines to be collected.
#   2.  Replace the target node list for each line with node names,
#       a file containing node names, or a ! to designate the
#       local node.  (See the splm reference page)
#   3.  Add to the node list or add additional collection commands
#       if needed.
#
# In this table the following target nodes need to edited:
#    allnodes       -   All nodes in the SP system
#    amd_server     -   amd server node (usually the cws)
#    cws            -   Control Workstation

#cws,allnodes: /etc/amd/amd-maps/*  /amd-maps/

#amd_server: /etc/exports etc_exports

#amd_server: /etc/filesystems etc_filesystems

#cws,allnodes: /etc/amd/amq > amq.output

#cws,allnodes: /bin/df > df.output

#cws: /var/adm/SPlogs/amd/amd.log
```

Figure 73. /spdata/sys1/amd.tab

6.2.6 Collecting System Data

Create service collections

From SMIT, the fast path to invoke for the Create Service Collections menu is:

```
smit spcreate_collect
```

In the following example, service collections are created using the table file ssp.tab. snap -gfk will be called to gather general system information (-g), file information (-f), and kernel information (-k). Additional log and system data designated in the ssp.tab table will be collected in the other directory created by snap. The nodename.tar.Z compressed tar file will be created.

```
splm -a service -t /spdata/sys1/logtables/ssp.tab -p 'gfk' -c
```

The fast path to invoke for the Remove Service Collections menu is:

```
smit spremove_collect
```

In the following example, we will remove the service collections created using the table ssp.tab:

```
splm -a service -t /spdata/sys1/logtables/ssp.tab -r
```

Gather service collections

The fast path to invoke for the Gather Service Collections menu is:

```
smit spgather_collect
```

The following example will show how the compressed tar files from the nodes in the table ssp.tab are gathered to the local node and written to the tape device /dev/rmt0. The -s option provides a method for gathering all the compressed tar files to tape or disk or mailing them without requiring disk space on the local node.

```
splm -a gather -k service -t /spdata/sys1/logtables/ssp.tab -d /tmp -s  \
-o /dev/rmt0
```

6.2.7 SP Error Log

It may be helpful in system problem determination to look at all error logs at once in parallel. It is not a good idea to copy the /var/adm/ras/errlog files from all the nodes to one central place and then run errpt against this central file, for the following reasons:

1. The time needed for copying is added to the sequential processing time of all nodes.
2. The total time required will be longer than that required for viewing the logs in parallel, and longer than error log processing requires for node information from the ODM database on each node.

Therefore it is better to use the dsh command in combination with the errpt command and options to view the error log. To establish this, you can perform the following:

1. View the summary information for all nodes to determine which one must be examined more closely:

   ```
   dsh -a errpt -s 0810050096 | pg
   ```

2. In the previous example, all error entries are listed that occurred after August 10, 1996 at 5 a.m. for every node defined in the SDR.
3. Select the nodes that have error entries that need more examination.
4. View the selected nodes, as in the following example:

   ```
   dsh -w sp21n1,sp21n4,sp21n10 errpt -a -s 0810050096 > /tmp/810errors
   ```

When the hardware supervisors indicate a warning or shutdown condition, the SP System Monitor writes a message using the BSD syslog facility and the AIX Error Log facility. For example, when the hardware supervisors determine that a fan has failed, the SP System Monitor writes a precise message into the log file that includes the time, node, type of error, variable name and, in some cases, associated values.

The installation process creates the default syslog file /var/adm/SPlogs/SPdaemon.log on the Control Workstation. You may want to configure your system to send the system log information to other locations. For example, you may want to send the SPdaemon.log messages to another

workstation for convenience. You can do this using the *@hostname* parameter in the /etc/syslog.conf file.

6.3 Error Notification Facility

Following is a description of the actions to be taken by the notification method EN_pend, located in the directory /spdata/sys1/err_methods. You can install this method to invoke pre- and post-action scripts and mail a report of the logged error. The script provides a suggested structure for notification methods and can be reused with different pre- and post-action scripts.

EN_pend method flow

1. EN_pend looks in the directory it resides in for a file with the same name and one with the *envs* suffix. An EN_pend.envs script is installed in the same directory. If found, it will set the environment variables EN_RUNDEFAULT and EN_MAILLOC.
2. EN_pend.envs sets the EN_RUNDEFAULT environment variable. It also sets the EN_MAILLOC variable to root at the Control Workstation (or, if this is impossible, to root at the local node).
3. EN_pend checks for a pre-action script in the same directory and name with a **.**pre suffix, and executes it if found.
4. EN_pend mails an expanded report of the error using the sequence of the error passed by the notification facility to the EN_MAILLOC. Note that this will only happen if the variables EN_RUNDEFAULT and EN_MAILLOC are set.
5. EN_pend checks for a post-action script in the same directory and name with a **.**post suffix, and executes it if found.

Installing a notification object

To add the EN_pend method to all nodes in the current partition so they will send a report whenever an error of type PEND (loss of availability of a device is imminent) occurs, enter the following commands:

```
penotify -a -n "PEND_err" -P -t "PEND" -m /spdata/sys1/err_methods/ \
EN_pend $1

penotify -a -n "pend_err" -P -t "pend" -m /spdata/sys1/err_methods/ \
EN_pend $1

penotify -a -n "Pend_err" -P -t "Pend" -m /spdata/sys1/err_methods/ \
EN_pend $1
```

Three objects are added with variations on PEND because uppercase is not always adhered to by all AIX LPPs and vendor functions. The $1 argument causes the Error Notification to pass the error sequence number to the notify method.

Create different pre- and post-action scripts

The EN_pend and EN_pend.envs scripts can be used to invoke different pre- and post-action scripts for different error events by creating links to them. EN_pend looks for .envs, .pre and .post scripts in the directory it is called from and by the same base name. In the following example, EN_pend is used for reporting

hdisk0 errors on the nodes sp21n7, sp21n8, sp21n9, and sp21n10, and performing the pre- and post-actions:

1. Create .pre and .post scripts on one of the nodes. For example, create EN_hdisk0.pre and EN_hdisk0.post in the directory /spdata/sys1/err_methods on node sp21n7.
2. Copy the .pre and .post files to the nodes sp21n8, sp21n9 and sp21n10 using the pcp command:

   ```
   pcp -w sp21n8,sp21n9,sp21n10 EN_hdisk0.pre  /spdata/sys1/err_methods/
   pcp -w sp21n8,sp21n9,sp21n10 EN_hdisk0.post /spdata/sys1/err_methods/
   ```

3. Create links to the EN_pend and EN_pend.envs scripts:

   ```
   dsh -w sp21n7,sp21n8,sp21n9,sp21n10 ln -s /spdata/sys1/err_methods/EN_pend
    /spdata/sys1/err_methods/EN_hdisk0
   dsh -w sp21n7,sp21n8,sp21n9,sp21n10 ln -s /spdata/sys1/err_methods/EN_pend.envs
    /spdata/sys1/err_methods/EN_hdisk0.envs
   ```

4. Add the notification object:

   ```
   penotify -w sp21n7,sp21n8,sp21n9,sp21n10 -f add -P -n "hdisk0_err"
   /spdata/sys1/err_methods/EN_hdisk0 $1 -N "hdisk0"
   ```

   To display the notification object, just created, enter:

   ```
   penotify -w sp21n7,sp21n8,sp21n9,sp21n10 -f show -n "hdisk0_err"
   ```

   To remove this notification, enter:

   ```
   penotify -w sp21n7,sp21n8,sp21n9,sp21n10 -f remove -n "hdisk0_err"
   ```

Important

Notification methods need to be accessible to each node that the notify object is added to. We recommend that the notification scripts are kept local on each node in case of network failure. File collections should be used to maintain updates to notification methods.

The following is an example of the /spdata/sys1/err_methods/EN_pend file:

```
#!/bin/ksh
#####################################################################
#                                                                   #
# Module: EN_pend                                                   #
#                                                                   #
#CPRY                                                               #
# 5765-296 (C) Copyright IBM Corporation 1995                       #
# Licensed Materials - Property of IBM                              #
# All rights reserved.                                              #
# US Government Users Restricted Rights -                           #
# Use, duplication or disclosure restricted by                      #
# GSA ADP Schedule Contract with IBM Corp.                          #
#CPRY                                                               #
#                                                                   #
#-------------------------------------------------------------------#
#                                                                   #
# Description: Default error notification script for pend errors.   #
#                                                                   #
# Syntax (example): Add as EN_pend $1                               #
#                                                                   #
# Internal Ref: None                                                #
#                                                                   #
#####################################################################
#@(#)96 1.1 src/ssp/logmgt/bin/EN_pend, sysman, ssp_r2.4, r2_4t6d6
4/4/95 07:21:32

unset EN_MAILLOC
unset EN_RUNDEFAULT

ENVS=$0'.envs';
PRE=$0'.pre';
POST=$0'.post';

if [ -x $ENVS ]
then
    .  $ENVS
fi

if [ -x $PRE ]
then
    echo $PRE;
fi

if [ "$EN_RUNDEFAULT" != "" ]
then
    if [ "$EN_MAILLOC" != "" ]
    then
        errpt -a -l $1 > /tmp/tmprpt.$$
        mail $EN_MAILLOC < /tmp/tmprpt.$$
        rm /tmp/tmprpt.$$
    fi
fi

if [ -x $POST ]
then
    echo $POST;
fi
```

Figure 74. /spdata/sys1/err_methods/EN_pend

The following is an example of the /spdata/sys1/err_methods/EN_pend.envs file:

```
#!/bin/ksh
#####################################################################
#                                                                   #
# Module: EN_pend.envs                                              #
#                                                                   #
#CPRY                                                               #
# 5765-296 (C) Copyright IBM Corporation 1995                       #
# Licensed Materials - Property of IBM                              #
# All rights reserved.                                              #
# US Government Users Restricted Rights -                           #
# Use, duplication or disclosure restricted by                      #
# GSA ADP Schedule Contract with IBM Corp.                          #
#CPRY                                                               #
#                                                                   #
#-------------------------------------------------------------------#
#                                                                   #
# Description: Sets up RUN_DEFAULT and MAILLOC env variables for    #
#              the EN_pend script.  Must reside in the same         #
#              directory as the EN_pend script to be called.        #
#                                                                   #
#####################################################################
#@(#)97 1.2 src/ssp/logmgt/bin/EN_pend.envs, sysman, ssp_r2.4, r2_4t6d6
6/5/95 09:01:29

#!/bin/ksh

export EN_RUNDEFAULT="YES";
export EN_MAILLOC=root

/usr/lpp/ssp/bin/SDRGetObjects SP control_workstation > /dev/null 2>&1
if [ "$?" = 0 ]
then
    CWS=/usr/lpp/ssp/bin/SDRGetObjects SP control_workstation  \
    | awk 'NR>1';
    if [ "$CWS" != "" ]
    then
        export EN_MAILLOC="root@$CWS";
    fi
fi
```

Figure 75. /spdata/sys1/err_methods/EN_pend.envs

6.3.1 Error Notification Objects

Error notification objects are ODM objects held in the class errnotify that are used by the AIX Error Notification facility to invoke methods upon occurrence of an error event.

Fields in the errnotify class match to fields in the Error Template for selection. If an error is logged matching the selection criteria defined in a notification object, the method associated with that object is invoked.

You can add, remove, and show notification objects in parallel on the RS/6000 SP system.

The fast path for the Add a Notification Object menu is:

```
smit padd_en
```

The fast path for the Remove a Notification Object menu is:

```
smit prem_en
```

The fast path for the Show a Notification Object menu is:

```
smit pshw_en
```

From the command line:

To add, remove, or show error notification objects in parallel on the SP system, enter:

```
penotify -f show
```

Useful options with the penotify command are:

-a	Executes on all nodes in the system partition.
-c	Specifies the error class.
-f	Specifies the function: show, add, remove.
-m *method*	Specifies the notification method.
-n *name*	Specifies the name of the notification object.
-w *hosts*	Run the command on a file or a list of hosts names separated by commas or blanks.
-P	This option will cause the object to persist after the system is restarted.

Note that you must be defined as a Kerberos principal in the /etc/logmgt.acl file to run this command.

To view all notification object on the nodes sp21n1, sp21n2, and sp21n3, enter:

```
penotify -w sp21n1,sp21n2,sp21n3 -f show
```

To remove the notification object named HDISK0_ERR on all nodes, enter:

```
penotify -a -f remove -n HDISK0_ERR
```

The AIX Error Notification facility can notify you when an SP error occurs. The error notification will perform an ODM method defined by the administrator when a particular error occurs or a particular process fails.

The following classifications of errors should have notifications defined by the administrator. Many of these messages will not occur often, so these notification objects should be defined even for large SP systems.

1. **PPS AIX Error Label that end in _EM**

   The _EM suffix signifies an emergency error and is usually used to tell the administrator some information that would be needed to reboot a node. You can find these messages by issuing the command:

   ```
   errpt -t | grep "_EM"
   ```

2. **Any AIX Error Log entries that have an Error Type of PEND**

   The error type PEND signifies an impending loss of availability and action will soon be required by the administrator.

3. **Any AIX Error Log entries for the boot device of the node**

   The boot device of the node usually is a resource of hdisk0, but the name may vary if the installation has been customized.

4. **The AIX Error Label EPOW_SUS**

   The EPOW_SUS error log entry is generated prior to a system power down when an unexpected loss of electrical power is encountered.

5. **The AIX Error Labels KERN_PANIC and DOUBLE_PANIC**

   KERN_PANIC and DOUBLE_PANIC error log entries are generated when a kernel panic occurs.

6.3.2 Error Notification Object Class

The error notification object class allows applications to be notified when particular errors are recorded in the system error log. The application describes the set of errors to be notified of in an Error Notification object.

Each time an error is logged, the error notification daemon determines if the error log entry matches the selection criteria of any of the error notification objects. If that is the case, then the daemon runs the notify method of each matched object.

The error notification object class is located in the /etc/objrepos/errnotify file. This object file contains the following descriptors:

en_pid Allows the owner of the error notification object to specify a process ID for use by the Notify Method. Objects which have a PID specified should have the en_persistenceflg set to 0.

en_name Uniquely identifies the object. The creator uses this unique name when removing the object.

en_persistenceflg Designates whether the error notification object should be automatically removed when the system is restarted. Valid values are:

- **0** Removes the error notification object at system restart.
- **non-zero** Retains the error notification objects at system restart.

en_crid Specifies the error identifier associated with a particular error.

en_label Specifies the label associated with a particular error identifier as defined in the header file /usr/include/sys/erridd.h.

en_class Identifies the class of the error log entries to match. Valid values are:

- **H** Hardware error class
- **S** Software error class
- **O** Messages from the errlogger command

en_type Identifies the severity of error log entries to match. Valid values are:

- **PEND** Impending loss of availability
- **PERM** Permanent
- **Perf** Unacceptable performance degradation
- **TEMP** Temporary
- **UNKN** Unknown

en_alertflg Identifies whether the error is alertable. This descriptor is provided for use by alert agents associated with network management applications. Valid values are:

TRUE Matches alertable errors.
FALSE Matches non-alertable errors.

en_resource Identifies the name of the failing resource. For the hardware error class, a valid resource name is the device name.

en_rtype Identifies the type of the failing resource. For the hardware error class, a valid resource type is the device type a resource is known by in the devices object.

en_rclass Identifies the class of the failing resource. For the hardware error class, a valid resource class is the device class. The resource error class is *not* applicable for the software error class.

en_method Specifies a shell script or command string to be run when an error matching the selection criteria of this error notification object is logged. The notification uses the `sh -c` command to execute the notify method. The following key words are automatically expanded by the error notification daemon as arguments to the notify method:

$1 The sequence number from the error log entry
$2 The error ID from the error log entry
$3 The error class from the error log entry
$4 The error type from the error log entry
$5 The alert flags from the error log entry
$6 The resource name from the error log entry
$7 The resource type from the error log entry
$8 The resource class from the error log entry
$9 The error label from the error log entry

Use the en_persistenceflg to avoid erroneous signaling. Those notification objects containing methods which send a signal to another process should especially *not* persist across system restart. This is because the receiving process and its process ID do not persist across system restarts. The creator of the error notification object is responsible for removing the error notification object at the appropriate time. In the event that the process terminates and fails to remove the error notification object, the en_persistenceflg descriptor ensures that obsolete error notification objects are removed when the system restarts.

6.3.3 Adding an Error Notification Object

The following example shows how to mail the error report to root@controlworkstation when a switch adapter fails the online diagnostics.

Adding the `dsh -a` command to the ODM commands will perform the action on all nodes of the RS/6000 SP.

1. Set up the directories for the error notification objects and methods:

   ```
   mkdir /<define_path>/errnotify/objects
   mkdir /<define_path>/errnotify/methods
   ```

2. Create the error notification method script.

   Create the script that will be run when the error occurs. For example:

```
#!/bin/ksh
#############################################################################
# Run errpt to get the full error report for the error that                 #
# was written and redirect it to a unique errnot.$$ file.                   #
# $$ will expand to the PID of this script.                                 #
#############################################################################

errpt -a -l $1 > /tmp/errnot.$$

#############################################################################
# Mail the full expanded error report to root@controlworkstation.           #
# This is the user and the hostname that the administrator wants            #
# to be notified at.  They could be anywhere in the system.                 #
#############################################################################

mail root@controlworkstation < /tmp/errnot.$$
```

3. Create the error notification object.

   Create a file that contains the error notification object to catch the tbx diagnostic failed error. It is easier to modify an existing set of ODM errnotify stanzas and edit the file. To do this, enter:

   ```
   odmget errnotify > /tmp/errnotify.odm
   ```

   Important

   Include only the attributes that have values.

   If you want to create a new file containing the error notification object, for example, tbx_diagerr.obj:, enter the following:

   ```
   errnotify:
            en_name = "tbx_diagerr.obj"
            en_persistenceflg = 1
            en_label = "HPS_DIAG_ERROR2_ER"
            en_method = "/<define_path>/methods/errnot.$1"
   ```

   Note that the en_name value can have a maximum length of 16 characters. Refer to the command odmshow errnotify to view the error notification object.

4. Add the error notification object to the errnotify class:

   ```
   odmadd /<define_path>/object/tbx_diagerr.obj
   ```

   To delete this object, enter:

   ```
   odmdelete -o errnotify -q "en_name = tbx_diagerr.obj"
   ```

   To view this object in the ODM database, enter:

   ```
   odmget -q "en_name = tbx_diagerr.obj" errnotify
   ```

5. The following mail will be sent to root@controlworkstation when a switch adapter fails diagnostics:

   ```
   From root@sp21n12.aixedu.nl.ibm.com Fri Aug 10 11:29:03 1996
   Received: from sp21n12.aixedu.nl.ibm.com by ppsras.aixedu.nl.ibm.com    \
             (AIX 4.1/UCB 5.64/4.03)
             id JT24554; Wed, 29 Mar 1996 15:30:42 -0400
   Date: Fri, 10 Aug 1996 11:29:03 -0400
   From: root
   Message-Id: <9608101525.JT24554@sp21n12.aixedu.nl.ibm.com>
   To: root
   Status: RO
   ```

```
---------------------------------------------------------------------------
ERROR LABEL: HPS_DIAG_ERROR2_ER
ERROR ID: 323C48A0

Date/Time:       Fri Aug 10 11:29:03
Sequence number: 18282
Machine Id:      000005911800
Node Id:         sp21n12
Error Class:     H
Error Type:      PERM
Resource Name:   Worm
Resource Class:  NONE
Resource Type:   NONE
Location:        NONE

Error Description
HPS adapter failed Online diagnostics

Probable Causes
Switch clock signal missing
HPS adapter disconnected

User Causes
Switch cable disconnected

Recommended actions
Run adapter diagnostics

Failure Causes
HPS adapter

Recommended Actions
Run adapter diagnostics

Detail Data
DETECTING MODULE
LPP=PSSP,Fn=pubstest.ksh,SID=1.0,Func=main,

Service Request Number
Testing Error Notify
```

Keep the method scripts on each node so you can run them if network or distributed file system problems occur. Using File Collections is an excellent way to keep these scripts updated. The object files may be in a distributed file system since they are not used unless changes to the object are required.

6.3.4 Mailing Error Reports to the Control Workstation

The following example shows how to mail the error report to root@controlworkstation when an error of TYPE PEND occurs.

Adding the dsh -a command to the ODM commands will perform the action on all nodes of the RS/6000 SP.

1. Set up the directories for the error notification objects and methods.

   ```
   mkdir /<define_path>/errnotify/objects
   mkdir /<define_path>/errnotify/methods
   ```

2. Create the error notification method script.

   Create the script that will be run when the error occurs. For example:

```
#!/bin/ksh
#############################################################################
# Run errpt to get the full error report for the error that                 #
# was written and redirect it to a unique errnot.$$ file.                   #
# $$ will expand to the PID of this script.                                 #
#############################################################################

errpt -a -l $1 > /tmp/errnot.$$

#############################################################################
# Mail the full expanded error report to root@controlworkstation.           #
# This is the user and the hostname that the administrator wants            #
# to be notified at.  They could be anywhere in the system.                 #
#############################################################################

mail root@controlworkstation < /tmp/errnot.$$
```

3. Create the error notification objects.

   Create the file that contains the error notification objects to catch the pending availability problems.

   Note that we have added some variations of PEND. This is because uppercase PEND is *not strictly* adhered to by all AIX LPPs and vendors.

```
errnotify:
         en_name = "errnot.PEND.obj"
         en_persistenceflg = 1
         en_label = "PEND"
         en_method = "/<define_path>/methods/errnot.$1"

errnotify:
         en_name = "errnot.Pend.obj"
         en_persistenceflg = 1
         en_label = "Pend"
         en_method = "/<define_path>/methods/errnot.$1"

errnotify:
         en_name = "errnot.pend.obj"
         en_persistenceflg = 1
         en_label = "pend"
         en_method = "/<define_path>/methods/errnot.$1"
```

4. Add the error notification objects to the errnotify class:

```
odmadd /<define_path>/object/errnot.pend.obj
```

   To delete these objects, enter:

```
odmdelete -o errnotify -q "en_name = errnot.PEND.obj"
odmdelete -o errnotify -q "en_name = errnot.Pend.obj"
odmdelete -o errnotify -q "en_name = errnot.pend.obj"
```

   To view these objects in the ODM database, enter:

```
odmget -q "en_name = errnot.PEND.obj" errnotify
odmget -q "en_name = errnot.Pend.obj" errnotify
odmget -q "en_name = errnot.pend.obj" errnotify
```

5. Mail will be sent to the administrator at the Control Workstation when an error of types PEND, Pend, or pend occurs.

6.3.5 Notification on Boot Device

The following example shows how to mail the error report to root@controlworkstation when an error on the boot device of hdisk0 occurs.

Adding the dsh -a command to the ODM commands will perform the action on all nodes of the RS/6000 SP.

1. Set up the directories for the error notification objects and methods:

   ```
   mkdir /<define_path>/errnotify/objects
   mkdir /<define_path>/errnotify/methods
   ```

2. Create the error notification method script.

   Create the script that will be run when the error occurs. For example:

   ```
   #!/bin/ksh
   ###############################################################################
   # Run errpt to get the full error report for the error that                   #
   # was written and redirect it to a unique errnot.$$ file.                     #
   # $$ will expand to the PID of this script.                                   #
   ###############################################################################

   errpt -a -l $1 > /tmp/errnot.$$

   ###############################################################################
   # Mail the full expanded error report to root@controlworkstation.             #
   # This is the user and the hostname that the administrator wants              #
   # to be notified at.  They could be anywhere in the system.                   #
   ###############################################################################

   mail root@controlworkstation < /tmp/errnot.$$
   ```

3. Create the error notification objects.

   Create the file that contains the error notification object to catch the boot disk error. In this example we assume that hdisk0 is the boot device:

   ```
   errnotify:
            en_name = "errnot.boot.obj"
            en_persistenceflg = 1
            en_label = "hdisk0"
            en_method = "/<define_path>/methods/errnot.$1"
   ```

4. Add the error notification object to the errnotify class:

   ```
   odmadd /<define_path>/object/errnot.boot.obj
   ```

   To delete this object, enter:

   ```
   odmdelete -o errnotify -q "en_name = errnot.boot.obj"
   ```

   To view these objects in the ODM database, enter:

   ```
   odmget -q "en_name = errnot.boot.obj" errnotify
   ```

5. Mail will be sent to the administrator at the Control Workstation when an error on hdisk0 occurs.

6.3.6 Notification Power Loss and PANIC

The following example shows how to mail the error report to root@controlworkstation when an unexpected power loss and kernel panic occur.

Adding the dsh -a command to the ODM commands will perform the action on all nodes of the RS/6000 SP.

1. Set up the directories for the error notification objects and methods:

   ```
   mkdir /<define_path>/errnotify/objects
   mkdir /<define_path>/errnotify/methods
   ```

2. Create the error notification method script.

   Create the script that will be run when the error occurs. For example:

   ```
   #!/bin/ksh
   ###############################################################################
   # Run errpt to get the full error report for the error that             #
   # was written and redirect it to a unique errnot.$$ file.               #
   # $$ will expand to the PID of this script.                             #
   ###############################################################################

   errpt -a -l $1 > /tmp/errnot.$$

   ###############################################################################
   # Mail the full expanded error report to root@controlworkstation.       #
   # This is the user and the hostname that the administrator wants        #
   # to be notified at.  They could be anywhere in the system.             #
   ###############################################################################

   mail root@controlworkstation < /tmp/errnot.$$
   ```

3. Create the error notification objects.

   Create the file that contains the error notification objects to catch the kernel panic and power loss error labels.

   ```
   errnotify:
            en_name = "power.obj"
            en_persistenceflg = 1
            en_label = "EPOW_SUS"
            en_method = "/<define_path>/methods/errnot.$1"

   errnotify:
            en_name = "panic.obj"
            en_persistenceflg = 1
            en_label = "KERNEL_PANIC"
            en_method = "/<define_path>/methods/errnot.$1"

   errnotify:
            en_name = "DBL_PANIC.obj"
            en_persistenceflg = 1
            en_label = "DOUBLE_PANIC"
            en_method = "/<define_path>/methods/errnot.$1"
   ```

4. Add the error notification object to the errnotify class:

   ```
   odmadd /<define_path>/object/errnot.POWER.PANIC.obj
   ```

   To delete these objects, enter:

```
odmdelete -o errnotify -q "en_name = power.obj"
odmdelete -o errnotify -q "en_name = panic.obj"
odmdelete -o errnotify -q "en_name = dbl_panic.obj"
```

To view these objects in the ODM database, enter:

```
odmget -q "en_name = power.obj" errnotify
odmget -q "en_name = panic.obj" errnotify
odmget -q "en_name = dbl_panic.obj" errnotify
```

5. Mail will be sent to the administrator at the Control Workstation when any power loss or kernel panic occurs.

6.4 Error Daemons

This section describes AIX error daemons and SP log daemons.

6.4.1 AIX Error Daemons

The error logging daemon errdemon reads error records from the special file /dev/error and creates error log entries in the system error log. Each time an error is logged, errdemon writes an entry in the system error log and performs error notify as specified in the error notification database /etc/objrepos/errnotify. The default system error log file is /var/adm/ras/errlog.

The last error log entry is always placed in NVRAM, and during system startup, this last error entry is read from NVRAM and added to the error log when errdemon is started.

errdemon does NOT create an error log entry for the logged error if the error record template specifies Log=FALSE.

If you use errdemon without any flags, the system will use the values stored in the error log configuration database for the error log file name, error log file size and internal buffer size.

Note that you need root authority to run errdemon.

The file /var/adm/ras/errtmplt contains the error template repository.

The file /etc/objrepos/SWservAt (software service aids attributes object class) contains the error log configuration database.

Flags

-i <file> (lowercase i). Uses the error log file specified by the <file> variable. The file name is saved in the error log configuration database and is immediately put into use.

-l (Lowercase L). Displays the values for the error log file name, file size and buffer size from the error log configuration database.

-s <logsize> Uses the size specified by the <logsize> variable for the maximum size of the error log file. The size limit is saved in the error log configuration database and is immediately put into use.

If the new size limit is smaller than the log file size currently used, errdemon renames the current log file to <name>.old and creates a new error log file with the specified size limit.

-B<buffsize> Uses the number of bytes specified by the <buffsize> parameter for the in-memory buffer used by the error log device driver. The specified buffer size is saved in the error log configuration database.

If the <buffsize> parameter is larger than the buffer size currently in use, then the in-memory buffer will be immediately increased.

If the <buffsize> parameter is smaller than the buffer size currently in use, then the new size is put into effect the next time errdemon is started after the system is rebooted.

The size you specify is always rounded up to the next integral multiple of the memory page size (4KB). The memory used for the error log device driver′s in-memory buffer is not available for use by other processes (the buffer is “pinned”).

Be careful not to impact your system′s performance by making the buffer excessively large. On the other hand, if you make the buffer too small, the buffer can become full if error entries arrive faster than they can be read from the buffer and put into the log file. When the buffer is full, new entries are discarded until space becomes available in the buffer. When this situation occurs, errdemon creates an error log entry to inform you about the problem. You can easily correct this problem by enlarging the buffer.

Note: On the RS/6000 SP system, thin and wide nodes behave differently regarding NVRAM and power loss.

You can use the command `errclear` to remove entries from the system error log.

Note that errdemon is normally started during system initialization. Stopping the error logging daemon can cause error data to be temporarily stored in internal buffers. These buffers can be overwritten before the error data is recorded in the error log file.

If you have specified a new error log file size and you want to generate an error report from the old one, you can use the `-i` option of the `errpt` command.

Note that the buffer cannot be made smaller than the hard-coded default of 8 KB.

Syslogd Daemon

The syslogd daemon reads a datagram socket and sends each message line to a destination described by the /etc/syslog.conf configuration file. The syslogd daemon reads the configuration file when it is activated and when it receives a hangup signal.

The syslogd daemon creates the /etc/syslog.pid file, which contains a single line with the command process ID used to end or configure the syslogd daemon. A terminate signal sent to the syslogd daemon ends the daemon, and syslogd will

log the end-signal information and terminate immediately. Each message is one line. A message can contain a priority code, marked by a digit enclosed in <> (angle braces) at the beginning of the line.

Configuration File

The configuration file informs the syslogd daemon where to send a message, depending on the message's priority level and facility. The syslogd daemon ignores blank lines and lines beginning with a # (pound sign).

Lines in the configuration file for the syslogd daemon contain a *selector* field and an *action* field, separated by one or more tabs. The *selector* field names a facility and a priority level. You can separate facility names with a comma (,) and separate the facility and priority-level portions of the *selector* field with a period (.). You can separate multiple entries in the same selector field with a semicolon (;). To select all facilities, you can use an asterisk (*).

The *action* field identifies a destination (file, host, user) to receive a message. If routed to a remote host, the remote system will handle the message as indicated in its own configuration file. To display messages on a user's terminal, the destination field must contain the name of a valid, logged-in user.

See Table 7 on page 190 for field definitions and uses.

Selector Field	Facility Names	**kern**	Kernel
		user	User Level
		mail	Mail subsystem
		daemon	System daemons
		auth	Security or authorization
		syslog	syslogd daemon
		lpr	Line-printer subsystem
		news	News subsystem
		uucp	uucp subsystem
		*****	All facilities
	Priority Levels	**emerg**	LOG_EMERG. Emergency (or panic) messages, for example, disk errors. Emergency/panic errors are not distributed to all users.
		alert	LOG_ALERT. Important messages such as a serious hardware error. These messages are distributed to all users.
		crit	LOG_CRIT Critical messages not classified as errors, for example, improper login attempts. LOG_CRIT and higher-priority messages are sent to the system console.
		err	LOG_ERR. Represents an error condition, for example, an unsuccessful disk write.
		warning	LOG_WARNING. Messages for abnormal, but recoverable conditions.
		notice	LOG_NOTICE. Important informational message. All messages without a priority are mapped to this priority.
		info	LOG_INFO. Informational messages. Normally you can discard these messages, but they can be useful in analyzing the system.
		debug	LOG_DEBUG. Debugging messages. You can discard these messages.
		none	Excludes the selected facility. This priority level is useful only if preceded by an entry with an * (asterisk) in the same selector field.
Action Field	Destination	**File name**	Full path name of a file opened in append mode
		@host	Host name, preceded by @ (at sign)
		User,user	User name(s)
		*****	All users

Table 7. Configuration File Field Definitions

6.4.2 SP Log Daemons

`setup_logd` sets up the SP logging daemon (splogd) and is called by installation scripts when the Control Workstation is installed. It can also be run by root on a different workstation to have splogd spawned by the SCR (System Resource Controller).

If you want to run the splogd logging daemon on a workstation other than the Control Workstation, you must install the *ssp.clients* option on that workstation and run `setup_logd`. Do this in the following order:

1. Off-load error logging from the Control Workstation.
2. Have your own script called when a state on a particular variable or variables occurs.

By default, the /spdata/sys1/spmon/hwevents file is set up to do error logging and state change logging for all frames. If you are installing splogd on a workstation besides the Control Workstation in order to call your own script, you must edit the /spdata/sys1/spmon/hwevents file and remove the entries for SP_STATE_LOG and SP_ERROR_LOG. You must also add a call for your own script. If you do not want to perform any of the following steps on your workstation, do not run `setup_logd`.

The `setup_logd` command performs the following steps:

1. It creates directories in /var/adm that the logging daemon uses. (Of course this will only be done if they do not already exist.)
2. It adds an entry to the file /etc/syslog.conf for the daemon.notice and sends a HUP signal to syslogd to reread its configuration file.
3. It adds errlog templates for SP messages.
4. It adds the splogd daemon to SRC as the splogd subsystem.
5. It adds an entry for splogd to /etc/inittab.

Important

If you are only using syslogd to call your own script, perform only Step 4 and Step 5: add syslogd to SRC and add an entry to /etc/inittab.

If you want to run the logging daemon on a separate workstation, you must add (or change) the following line to the /etc/environment file:

```
SP_NAME=<Control Workstation>
```

If you want to move a subset of error logging off the Control Workstation, you must edit /spdata/sys1/spmon/hwevents on the Control Workstation and define the new subset that you want to monitor. Then issue `stopsrc` command to stop the logging daemon on the Control Workstation, and issue `startscr` reread the hwevents file at start time.

The SP logging daemon splogd has the following functions:

error logging Reports the SP hardware errors to both the syslog and the AIX error log.

state change logging Writes SP hardware state changes to a file.

user exits Calls a user exit when a state change occurs.

The hwevents file contains state change actions that are to be performed by the splogd logging daemon. The fields in this file are:

frame Specifies the frame number (1-*n*) or * (asterisk) for all frames.

slot Specifies the following:

- A number from 0-17
- One of
  - NODES_ONLY (addresses 1 through 16)
  - SWITCH (address 17)
  - FRAME (address 0)
  - * (asterisk, all addresses)
  - NODES_AND_SWITCH (addresses 1-17)
  - FRAME_AND_NODES (addresses 0-16)
  - FRAME_AND_SWITCH (addresses 0 and-17)

variable Specifies a hardware variable. For example, nodePower, Temp, LED7SegA.

operator Specifies how to compare the value. Acceptable values are: =, <, > and !=.

value Specifies the value of the variable to match the operator wildcard (*) or a partial match with the wildcard at the end (23*).

time Specifies if the function should be called at startup or when the state changes, or both times. Valid options are **startup, change** or **both**.

function Specifies the program to call when the event occurs. There two special keywords for **function** If **function** is SP_ERROR_LOG, error logging is performed provided that syslog is set up and AIX error logging is set up to perform SP logging. Refer to the setup_logd.

If **function** is SP_STATE_LOG, these state changes that meet the statement's criteria are logged to the file /var/adm/SPlogs/spmon/splogd.state_changes.<timestamp>.

Note that you must send a SIGHUP signal to splogd to close the current state_changes.<timestamp> and to open a new one. For example:

```
kill -HUP <splogd_PID>
```

User Exit Arguments

When a user exit is called by splogd, the following arguments are passed:

1. A **c** or **s** depending on whether this call is for a change of state or to provide the startup values for the variables being monitored.
2. For each variable being reported, the following arguments are passed:
   - Frame number
   - Node number
   - Variable name
   - Value of the variable. (Boolean values are expressed as TRUE or FALSE, integers as decimal strings, and floating-point values as floating-point strings.)

Starting and Stopping the splogd Daemon

The splogd daemon is under System Resource Controller (SRC) control. It uses the signal method of communication in the SRC. The splogd daemon is a single subsystem and not associated with any SCR group. The subsystem name is splogd.

In order to start the splogd daemon, use the `startsrc -s splogd` command. This command starts the daemon with the default arguments and SRC options. The splogd daemon is set up to be respawnable and to be the only instance of the splogd daemon running on that particular node or Control Workstation.

Important

Do *not* start splogd from the command line without using the `startsrc` command to start it.

To stop the splogd daemon, you can use the command `stopsrc -s splogd`. This will stop the daemon and does not allow it to respawn.

To display the status of the splogd daemon, issue the command `lsscr -s splogd`.

You can view the current SRC options and daemon arguments with the command:

```
odmget -q "subsysname=splogd" SRCsubsys
```

You can change the default startup arguments with the command `chssys`.

Chapter 7. Isolating Problems on the SP System

Most of the problems covered in previous chapters can be grouped into four main categories. Each of those categories has different types of errors.

This chapter provides some reference material, in flowcharts and in overviews, that gives pointers on how to resolve SP problems in those categories. These flowcharts and overview charts are not complete; they only present well-known problems and some hints that can help to identify a problem.

7.1 Isolating Booting Problems

The following sections discuss network booting considerations and problems.

7.1.1 Booting Process Overview

In order to network boot and install nodes, they must be defined as NIM clients for the NIM master (in NIM jargon), or boot/install server (in SP jargon). When they are defined, the NIM master allocates the necessary resources for them to boot and install.

The NIM database contains the NIM objects that define clients and resources, along with attributes and states for those objects. When setup_server is run, after set the boot_responds to install or customize for the affected nodes, this script executes several NIM commands for allocating resources to each client.

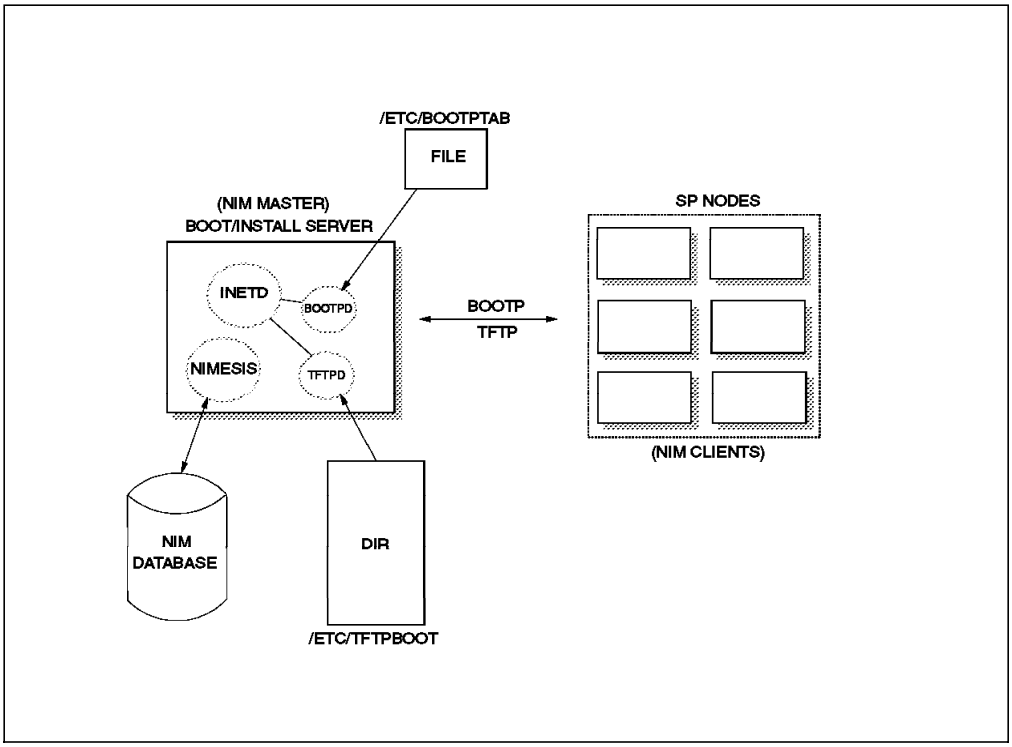


Figure 76. Booting Process Overview

The most visible and easily-checked results that happen when resources have been allocated to clients to boot and install are the /etc/bootptab file and the

/etc/tftpboot directory. They keep the information and resources to network boot or customize the nodes.

More information about this process and what files are created, can be found in Chapter 2, “The Installation Process” on page 7.

7.1.2 Booting Problems

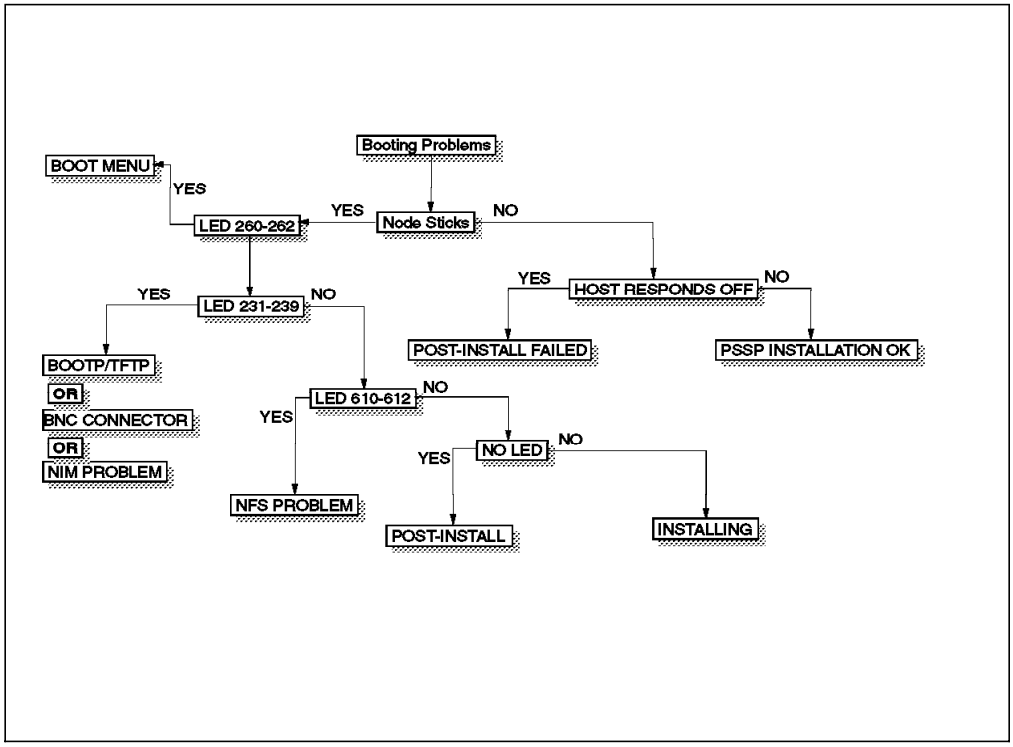


Figure 77. Booting Problems

The booting process has several components that work together to accomplish the tasks of network booting, installing, and customizing a node. During the first steps of this process, you have to specify who will serve the nodes you want to install. Along with that, you set the boot responds to install or customize in the SDR. Finally, you run setup_server. This script executes several NIM commands to allocate the necessary NIM resources to each client. Note that if you are installing several nodes with several NIM masters, setup_server will run in each one of the NIM masters that will be part of this installation or customization.

If you encounter problems running setup_server, look at 2.9.2, “Setup_server Fails” on page 56, which covers those kinds of problems. However, after running setup_server successfully, you are about to start the booting process.

To start installing or customizing your nodes, you must reboot them or power them on, if they were powered off. When they start booting, you will see the LED in the panel display.

The LED numbers are very useful to determine which step is being executed. In the chart, you see LED numbers indicating where nodes usually get stuck, for several reasons.

LED 260-262 These numbers indicate that the node is displaying the boot menu but there must be something wrong with the serial link, because nodecond cannot get messages from there.

LED 231-239 This is a booting problem. The node is intending to boot from the network, but no one is responding. It could be that the inetd is not responding, or the bootp or tftp daemons are not responding. Also, it can be a NIM problem, or an inconsistency between NIM and the SDR.

LED 610-612 This means that the node was unable to mount a remote resource using NFS. It is difficult to determine which resource mount failed.

- On the NIM master, verify that NFS is functioning correctly and that the appropriate resources have been exported correctly.
- If you have a gateway between the NIM client and NIM server, verify that the route defined in the NIM configuration is correct.

If you do not find any of the problems mentioned above, a good solution would be reset the NIM client and start all over again.

The final step of the installation process is customization. This is done by the pssp_script script. Many problems can be found if you look at the console log file for the particular node.

7.2 Isolating System Monitor Problems

The following section discusses how to isolate system monitor problems.

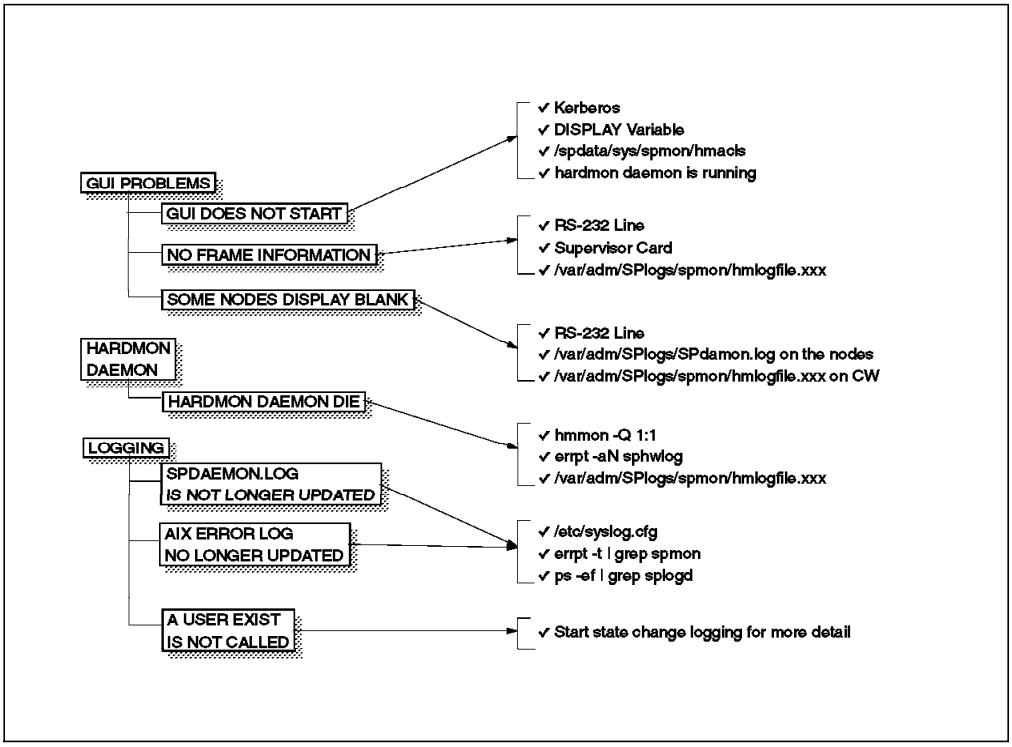


Figure 78. System Monitor Problems

Problems with System Monitor GUI

- spmon command fails to start the System Monitor GUI. Then check the following:
  - Verify authorization
    - Run klist command to verify that kerberos tickets have not expired.

      Important

      spmon will not run if kerberos tickets never expire.

    - Re-issue kinit command, if required.
    - Check the /spdata/sys1/spmon/hmacls file to ensure that the kerberos principal name and instance for your ID is in this hardware monitor ACL file.

```
sp2cw0 root.admin a
sp2cw0 hardmon.sp2cw0 a
1 root.admin vsm
1 hardmon.sp2cw0 vsm
```

  - Ensure you have exported the DISPLAY variable:

```
# echo $DISPLAY
<hoatname or IP address>:0
```

- If the open session failure message window is displayed after opening the System Monitor GUI, then verify authorization using klist and request authorization using the kinit command. If the problem persists, check the /spdata/sys1/spmon/hmacls file.

Hardware monitor daemon (hardmon) is not running

- Execute the following command to verify that the hardmon daemon is functioning correctly:

```
# hmmon -Q 1:1
```

- If the above command does not work, check the hardmon log file at /var/adm/SPlogs/spmon/hmlogfile.xxx, where xxx is the creation date.

Logging problems

- Logfile is no longer being updated

  Check the logging daemon

  - Check that splogd is running

    The following example shows how you can check that the splogd is running:

```
# lssrc -a |  grep splogd
 splogd                           15378   active

or,

# ps -eaf |  grep splogd
    root 15378  5854 0   May 08  -  0:00 /usr/lpp/ssp/bin/splogd
-f /spdata/sys1/spmon/hwevents
    root 18408 15248 0 11:22:43  pts/0 0:00 grep splogd
```

– Check that the following entry is in the /etc/syslog.conf configuration file:

```
daemon.notice          /var/adm/SPlogs/SPdaemon.log
```

– Check that the /var/adm/SPlogs/SPdaemon.log file has the correct permissions (rw-r--r--).

– Check that the error record templates for SPMON exists:

```
# errpt -t | grep SPMON
```

7.3 Isolating Switch Problems

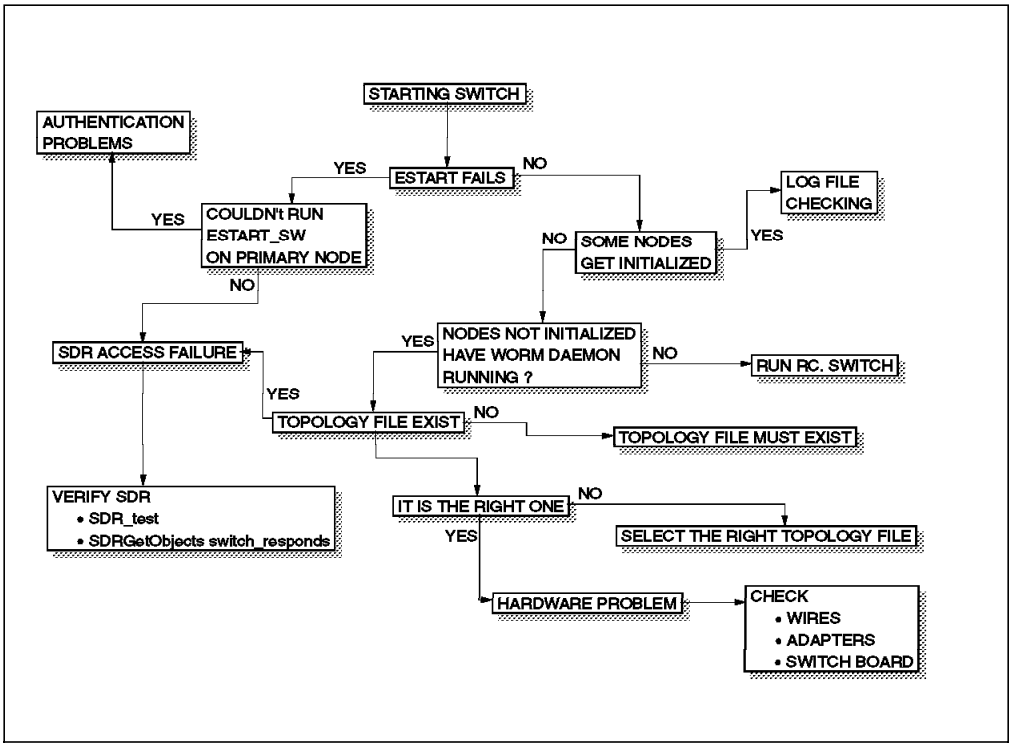


Figure 79. Switch Problems

The SP switch problems are usually related to four main components:

- **Estart**

  This command is executed when you want to start the switch. However this command is not the real one; the real one resides in the primary node and is called Estart_sw. Chapter 4, “The Switch” on page 87 describes all the tasks accomplished by these commands, so you can check there if you face problems related to them.

- **Switch Processes**

  These processes must be running in all the nodes participating in the switch network. So, always check that those processes are running in each node. The rc.switch script starts those processes and it can be executed multiple times if necessary.

- **Topology and clock files**

These files define how the nodes will communicate. The topology files define the links between nodes, and the clock files define how the clock is sourced into each switch board and adapter. If your topology or clock file is wrong, you will not be able to use the switch. Always check that the SDR contains the right files.

- **Hardware**

  Because the switch is a very stable SP component, hardware problems are in the common sources of switch-related problems.

7.4 Isolating System Partitioning Problems

As we discussed in Chapter 6, “Error Logging” on page 159, many PSSP components get partitioned when a system partition is applied. Nevertheless, many components remain systemwide.

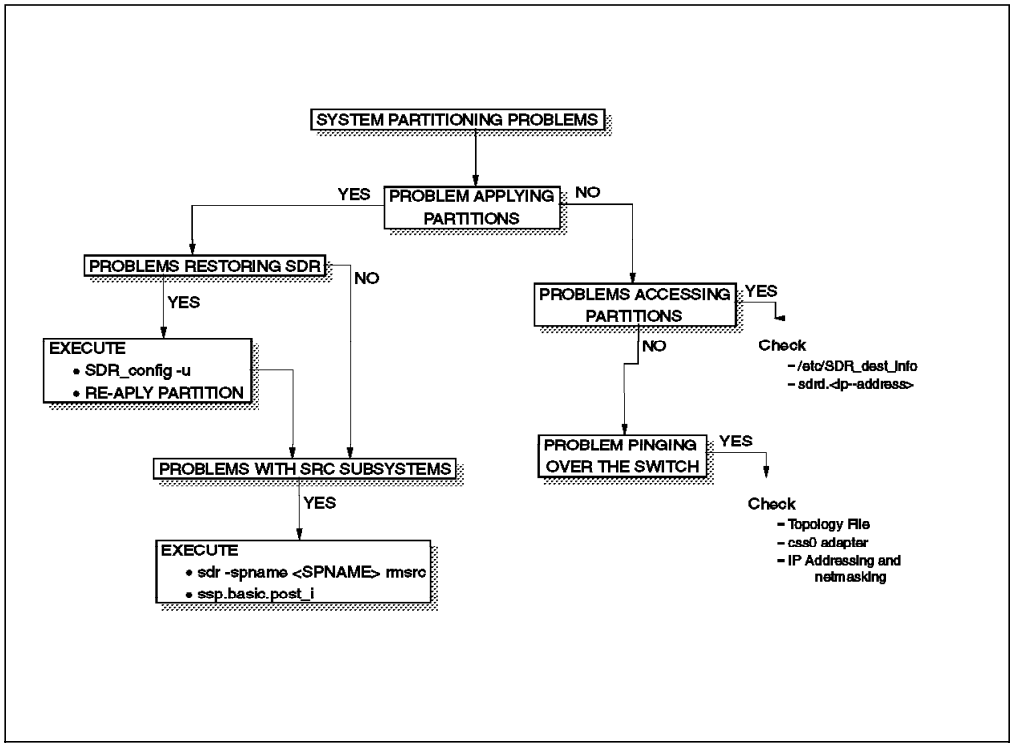


Figure 80. System Partitioning Problems

System partitioning problems usually arise from two sides:

- **Problems applying system partitions**

  There several ways to recover a failed system partition apply. These ways usually consist of recovering the SDR (previously saved). If that fails, you still have a chance to recover your system by executing some SDR commands that were described in Chapter 5, “System Partitioning” on page 125.

- **Problems accessing partitions**

  Usually these problems are related to missing topology files or problems with the SDR. Remember that several SP components also get partitioned when you apply a system partition, so be careful to check that your commands are querying to the right set of daemons.

Chapter 8. Producing a System Dump

There is not much difference in the way system dumps are handled in an SP environment. The only exception to this is in the primary dump device used by the nodes when they are running AIX Version 4.1.

- In AIX Version 3.2.5, the default dump device was /dev/hd7/, a special device for system dumping.
- With AIX Version 4.1, the primary dump device was changed to /dev/hd6, which is the default paging space.

  This way, you will not need a special device with additional disk space to copy a system dump when this is generated.

However, in the SP nodes, the special device /dev/hd7/ is still used, even when the nodes are running AIX Version 4.1, because the nodes do not have a console directly attached, and after a system dump is generated, the system intends to copy it to the /var/adm/ras directory.

Therefore, when you are using the paging space, this system dump must be copied; otherwise it will get lost when the system starts.

If you do not have space on the /var file system, the system will display a menu on the console asking for an answer about what to do with the system dump. This will cause your node to hang while it waits for an answer from the serial link.

This chapter gives a brief overview of system dumps and provides examples with commands and scripts you can use to handle them.

8.1 Handling Systems Dumps

The system generates a system dump when a severe error occurs. System dumps can also be user-initiated by users with *root* authority. A system dump creates a picture of your system's memory contents. System administrators and programmers can generate a dump and analyze its contents when debugging new applications.

When the system halts with a flashing *888* followed by a *102*, then you know that a system dump has occurred. When this happens, you must reboot the system in order to get the system back again.

There are two types of system dumps.

System-initiated	Produced if a kernel panic occurs.
User-initiated	Generated by a command or by using the key lock and the reset button.

A system-initiated dump is a snapshot of the kernel. A system dump occurs when a kernel panic takes place. This dump includes selected kernel structures, as specified in the dump table. The dump table is a kernel structure containing a list of structures that have been loaded, such as device drivers and program code. This table is dynamic, so there is definite rule as to how large the dump will be. The dump does not include paged memory.

Following are some of the kernel structures that are captured by the system dump:

System Variables and Statistic

This comprises the kernel parameters either set by the user or hardcoded into the kernel. Such statistics include *maxuproc* (maximum number of processes per user), *fsize* (maximum file size), and *stack* (maximum stack segment size).

Process Table

The process table contains a list of all currently running processes and relevant information about those processes.

User Areas

This includes more information about the current running processes on the system. It includes the file descriptor table.

VFS Information

This includes information about currently mounted file systems, the inode table and the open file table.

Kernel Stack

The kernel stack keeps the track of all the processes that were running in kernel mode at that time.

System Buffers

This is the area where incoming data is stored while awaiting memory allocation from the kernel.

TTY information

Contains information about currently configured ttys, their characteristics, and current state (including line discipline).

Mbufs

These are the memory buffers for data which have been sent/received across the network.

Sockets

A socket is a logical device which can be described as the end of a pipeline by which network data is transmitted.

When an unexpected system halt occurs, the system dump facility automatically copies selected areas of the kernel data to the primary dump device. These areas include kernel segment 0 as well as other areas registered in the master dump table by kernel modules or kernel extensions. Memory-resident user data is also dumped.

AIX version 4 maintains two system dump devices:

Primary Usually used when you wish to save the dump data.

Secondary Can be used to discard dump data (for example, /dev/sysdumpnull).

Make sure you know your system and know what your primary and secondary dump devices are set to.

- To list the current dump destination:

```
# sysdumpdev -l
  primary               /dev/hd6
  secondary             /dev/sysdumpnull
  copy directory        /var/adm/ras
  forced copy flag      TRUE
  always allow dump     FALSE
```

- To set the dump device:

```
# sysdumpdev [-P]  {-p|-s} <device>
```

Dump devices are usually portable mediums such as tape drives. However, for the SP nodes, those tape drives are not attached directly. You may copy the content of the dump into a file, then transfer it to the Control Workstation, and then download it to a tape.

A dump device can be configured either as a dedicated device that requires no operator intervention when a dump occurs, or as a shared device that can be used for other purposes until a dump is requested. A dedicated dump device is referred to as the primary dump device, and a shared one as a secondary dump device.

Operator intervention is required when a dump is to be written to the secondary dump device. When you install the operating system, the dump device is automatically configured for you. By default the primary device is /dev/hd6, which is a paging logical volume, and the secondary device is /dev/sysdumpnull. Note that for systems which have been migrated from AIX Version 3.2.5 to AIX Version 4.1, the primary dump device is set to what it was formerly /dev/hd7.

Important

/dev/hd7 is the primary dump device by default on the SP nodes, even in version 4.1. The reason for this is simple: Because /dev/hd6 is the paging space, after a system dump the node comes back and waits in the S1 line for a response about what to do with the system dump that resides in its paging space.

Use the sysdumpdev command or SMIT to query or change the primary and secondary dump devices.

- To estimate the size of the dump for the current system:

```
# sysdumpdev -e
  estimated dump size in bytes: 16250880
```

- To display statistical information about the previous dump:

```
# sysdumpdev -L
```

Flags:

-l List

-p Primary

-s Secondary

-P Make change permanent

-d directory Specifies the directory where the dump is copied at boot time. If the copy fails, the system continues to boot.

-D directory Specifies the directory where the dump device is copied at boot time. If the copy fails, then a menu is displayed to allow the user to copy the dump.

-z Writes out to standard output the string containing the size of the dump in bytes and the name of the dump device, if a new dump is present.

AIX version 4 uses /dev/hd6 (paging) as the default dump device unless the system was *migrated* from AIX version 3, in which case it will continue to use the AIX version 3 dump device /dev/hd7.

Important

The previous paragraph does not apply for the SP nodes, which use /dev/hd7 as the default primary dump device.

If a dump occurs to paging, the system will automatically copy the dump when the system is rebooted. By default, the dump gets copied to the /var/adm/ras directory.

The recommended value is that the dump device is at least a quarter of the size of real memory. In such situations, it is advisable to create a temporary dump logical volume of the size required and manually recreate the environment in which the dump occurred. If the dump device is not large enough, the system will produce a partial dump only. It is possible, but extremely unlikely, that a Support Center can determine the cause of the crash from a partial dump. The *-e* flag can be used as a starting point to determine how big the dump device should be.

8.1.1 How to Start a Dump

There are three ways for a user to invoke a system dump, but only two are valid on the SP nodes, since keyboard input is not possible. Which method is used depends on the condition of the system.

If there is a kernel panic, the system will automatically dump the contents of real memory to the primary dump device.

The user/administrator can initiate a system dump at any time by pressing the reset button (using spmon or GUI) while the keylock position is set to service mode.

Bear in mind that if your system is still operational, a dump taken at this time will not assist in problem determination. A relevant dump is one taken at the time of the system halt.

Do not start a dump if the flashing *888* number shows on the LED. This number could indicate that a dump has already occurred on your system. (You can determine this by finding out the LED code that is displayed after the flashing *888*. If it is a *102*, then this indicates that a dump has occurred). This indicates that your system has already created a system dump and has written the

information to the primary dump device. If you start your own dump before copying the information in your dump device, your new dump will overwrite the existing information.

A user-initiated dump is different from a dump initiated by an unexpected system halt because the user can designate which dump device to use. When the system halts unexpectedly, a system dump is initiated automatically to the primary dump device.

You can start a dump using one of the following methods:

- **Command Line**

  This method uses the `sysdumpstart` command. Note, however, this command is only available if you install the Software Service Aids (bos.sysmgt.serv_aid) package.

  You must have *root* authority to run this command. First you might want to check the current settings of your system dump devices by using the `sysdumpdev -l` command. Then initiate the dump with `sysdumpstart -p` (for primary device) or `-s` (for secondary device). Note that if the LED display is blank, the dump was not started. Try again using a different method.

- **Using SMIT**

  You can use the fast path: `smity dump`.

- **Using special key sequence**

  This option is only valid for the Control Workstation. You can initiate a dump, either to the primary or secondary dump device, by using the following key sequence:

  ```
  <crtl-alt-NUMPAD1>
  ```

- **Using the reset button**

  This procedure works for all system configurations and will work in circumstances where other methods for starting a dump will not. Turn the key into service position and press the reset button once. The system writes the dump information to the primary dump device.

If a system dump is initiated by kernel panic, the LEDs will display *0c9* while the dump is in progress, and then flash *888*. Using the reset button to rotate through the other LED codes, you will eventually encounter one of the codes shown in the foil. The code you want is *0c0*, indicating that the dump completed successfully. All of the LED codes following the flashing *888* should be recorded.

For user-initiated system dumps to the primary dump device, the LED codes should indicate *0c2* for a short period, followed by *0c0* upon completion.

Other common codes include:

0c4 This indicates that the dump routine ran out of space on the specified dump device. It may still be possible to examine and use the data on the dump device, but this tells you that you should increase the size of your dump device.

0c5 This indicates you should check the availability of the medium to which you are writing the dump.

0c7 This indicates a network dump is in progress, and the host is waiting for the server to respond. The value in the three-digit display should alternate between *0c7* and *0c2* or *0c9*. If the value does not change, then the dump did not complete due to an unexpected error.

0c8 This indicates you have not defined a primary or secondary dump device. The system dump option is not available. Enter the sysdumpdev command to configure the dump device.

0c9 This indicates a dump started by the system did not complete. Wait for one minute for the dump to complete and for the three-digit display to change. If the three-digit display value changes, find the new value on the list. If the value does not change, then the dump did not complete due to an unexpected error.

8.1.2 Copying a System Dump

The following section discusses details of copying a system dump.

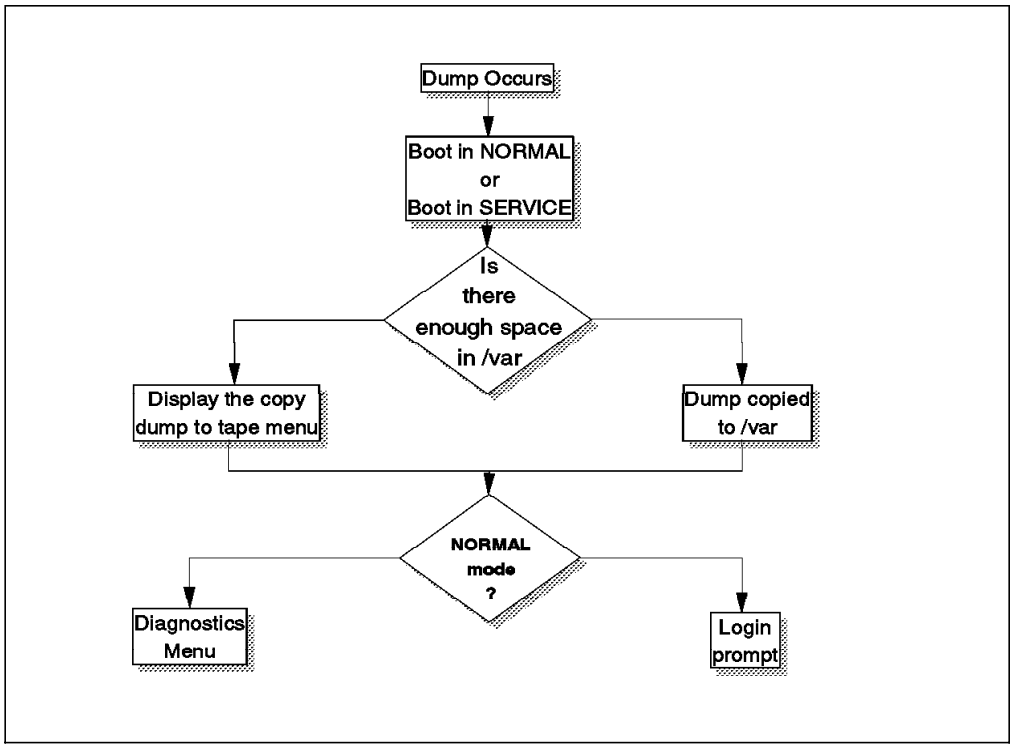


Figure 81. Copying a System Dump

After a crash, if the LED displays *0c0*, then you know that a dump occurred and it completed successfully. Press the reset button once. If there is enough space to copy the dump from paging to the /var/adm/ras directory, then it will be copied directly. If, however, at bootup the system determines that there is not enough space to copy the dump to /var, the /sbin/rc.boot script (which is executed at bootup) will call on the /lib/boot/srvboot script. This script in turn calls on the copydumpmenu command, which is responsible for displaying the following menu which can be used to copy the dump to removable media.

```
                Copy a System Dump to Removable Media

    The system dump is 16957952 bytes and will be copied from /dev/hd6
    to media inserted into the device from the list below.

    Please make sure that you have sufficient blank, formatted
    media before you continue.

    Step One:  Insert blank media into the chosen device.
    Step Two:  Type the number for that device and press Enter.

           Device Type                   Path Name

>>>




    88  Help ?
    99  Previous Menu

>>> Choice[0]:
```

Note: You will need to attach an external device to the SP nodes in order to use this option.

To copy a system dump after rebooting in normal mode, do the following:

- If it is the Control Workstation, log in as *root* and copy the system dump to either diskette or tape using snap command:

  ```
  # /usr/sbin/snap -gfkD -o /dev/rfd0
  ```

  (The example shown copies the dump to diskette)

  If the dump went to the paging space logical volume, it has been copied to the directory /var/adm/ras in rootvg. These dumps are also copied by the snap command.

- If it is a node, you may use the following procedure:

```
#!/usr/bin/ksh
#
# Usage: mkdump [filename]  [block size]
#
# Description:
#    Copy the system dump from the primary dump device to
#    file name (/var/adm/ras/dump_file by default) using the
#    block size specified (4096 by default).
#
dev=sysdumpdev -l|awk '/primary/ {print $2}'
size=sysdumpdev -L|awk '/Size/ {print $2}'
dir=sysdumpdev -l|awk '/directory/ {print $3}'

bs=${2:-4096}
file=${1:-dump_file}

count=expr $size / $bs
OVFL=expr $size % $bs
if · $OVFL != 0 "
then
   count=expr $count + 1
fi

echo "Copying from $dev to $file $size bytes..."
echo "....please wait...."
dd if=$dev of=$dir/$file bs=$bs count=$count
```

8.1.3 Sending the Dump to the Support Center

If a system halt occurs once, there is probably little point in sending the dump for analysis. However, if the system is halting repeatedly and this is seriously impacting the use of the machine, the dump can be sent to the Support Center (through your normal first-level support channel) for analysis.

The Support Center will analyze the contents of the dump using the crash command. The crash command uses the kernel that was active on the system at the time of the halt. Therefore, /unix should be sent from the machine where the dump was produced.

The AIX command snap was developed by the support team in Austin to simplify gathering configuration information. It provides a convenient method of sending the *lslpp* and *errpt* output to the Support Center. It gathers system configuration information and compresses the information to a tar file. The file can then be downloaded to disk, or tape.

Some useful flags with the snap command are:

-f Gather file system information

-g Gather general information

-k Gather kernel information

-D Gather dump and /unix

The information gathered with the `snap` command can be used to identify and resolve system problems. You must have *root* authority to execute this command.

If you use the *-a* flag, then you need approximately 8 MB of temporary disk space to collect all the system information, including the content of the error log.

The *-g* flag gathers the following information:

- Error report
- Copy of the customized ODM
- Trace file
- User environment
- Amount of physical memory and paging space
- Devices and attribute information
- Security user information

The output from the *-g* flag is written to /tmp/ibmsupt/general/general.snapfile. However, you can specify another directory using the *-d* flag.

The execution of the `snap` appends information to the previously-created files. Use the *-r* flag to remove previously-gathered and saved information.

Before you send your media to the Support Center, make sure you contact the Center and obtain a Problem Management Number (PMR) which will be used to trace the status of your problem from then on. Make sure you label the media with this number and also the other pieces of information listed, to help the support team act quickly on your problem.

There is not much left for you to do after this, apart from waiting for a response from the Support Center. However, you may want to look at your dump to try to analyze it yourself. The tool that is used by the Support Center to analyze your dump is called `crash` which also is available on the system. However, the output from the command is very user-unfriendly, and most people do not bother to analyze it. However, the following section describes how to use `crash` to analyze a system dump.

8.1.4 Using crash to Analyze a Dump

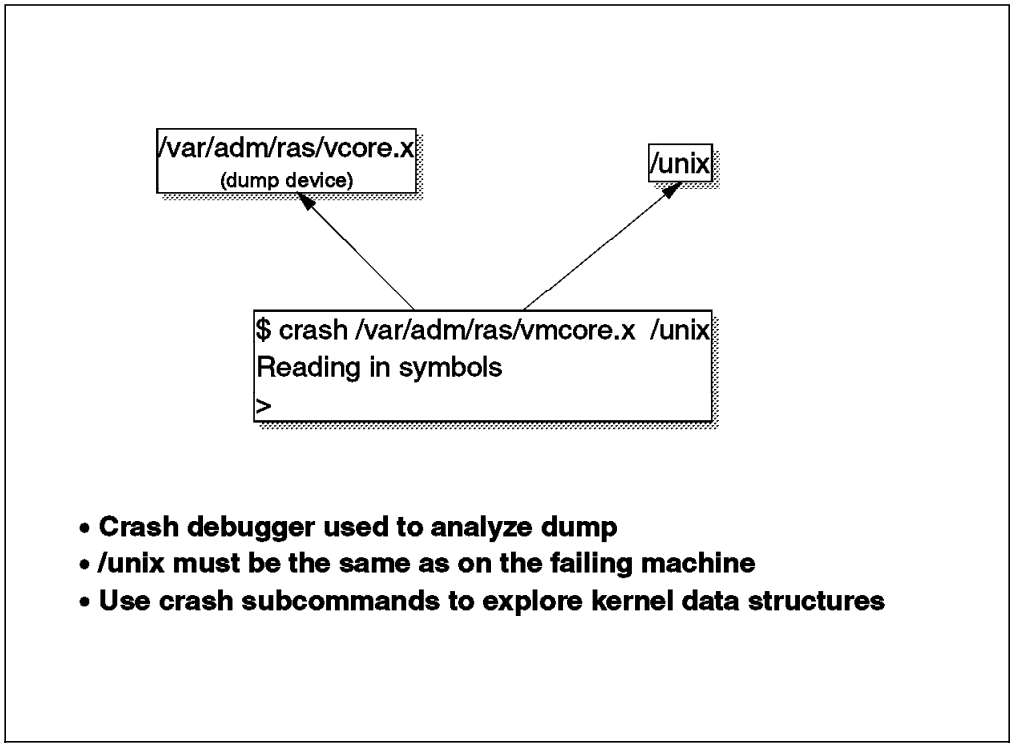


Figure 82. Using crash

The dump can be analyzed by using the crash command. The crash command can be used to check the validity of the dump:

```
# crash /var/adm/ras/vmcore.x /unix
```

If the following appears on the screen, then the dump is good:

```
Reading in symbols
>
```

If the copy of /unix does not match the dump file, the following output will appear on the screen:

```
WARNING:  dump file does not appear to match namelist
Reading in symbols
>
```

If the dump itself is corrupted in some way, then the following will appear on the screen:

```
0452-179 Cannot read vstructure from address
>
```

To exit the crash debug program, type *quit* at the > prompt.

From the > prompt, you can enter a number of crash subcommands. The following example shows typical values that might be seen.

The crash command and subcommands are fully documented in InfoExplorer, but the following shows basic examples of the use of crash subcommands:

```
> stat  (to show machine status at time of crash)
        sysname:  AIX
        nodename:  sp21n02
        release:  1
        version:  4
        machine:  000168205700
        time of crash:  Tue Aug  6 15:38:38 1996
        age of system:  7hr, 22min
        abend code:  0
        csa:  0x0
> status
CPU     TID  TSLOT     PID  PSLOT  PROC_NAME
  0    2b43     43    2e3a     46  sysdumpstart
```

The status subcommand displays a description of the kernel process. The information includes CPU number, thread ID, thread table slot, process ID, process table slot and process name. In our example, sysdumpstart is in process slot 46. To see further information about this process:

```
> p -r
SLT ST    PID   PPID   PGRP   UID  EUID  TCNT  NAME
  2 a     204      0      0     0     0     1  wait
        FLAGS: swapped_in no_swap fixed_pri kproc
 26 a    1aee   2da4   1aee     0     0     1  find
        FLAGS: swapped_in
 46 a    2e3a   2da4   2e3a     0     0     1  sysdumpstart
        FLAGS: swapped_in
0452-1004: Cannot read process table entry 37.
```

(Ignore messages reading “cannot read process table entry.”)

This shows that the process running at the time of the crash was sysdumpstart and it was running in slot 46. It also shows find was running in slot 26. To see the stack trace for find:

```
> t 26
STACK TRACE:
        .et_wait()
        .poll_wait()
        .select()
        .sys_call()
```

The crash command is an interactive utility for examining an operating system image, a core image, or the running kernel. It also interprets and formats control structures in the system and certain miscellaneous functions for examining a dump.

In order to analyze the dump, you must execute the crash command against /unix, and it must be the /unix of the system that had the problem. However, to make any change to code, you must have the source AIX code which is not held by customers. Therefore, there is not much more that you can do. It is best to let the Support Center handle the dump.

One useful operation you can do is to make sure that you have a good full dump. If you see :

```
Reading in symbols
>
```

after executing the crash command, you know that you have a full dump that can be analyzed. You should avoid sending dumps to the Support Center, only to find out that the Center cannot do anything about them because they are partial dumps. Get it right from the start.

Chapter 9. User and Services Management

This redbook is not intended to cover system management issues. The problem determination procedures described in this book are related to basic PSSP components, and are mainly operational tasks performed by the system administrator.

User and Services Management tasks are covered in other publications.

This chapter gives basic hints as to how to deal with certain problems related to SP system management.

9.1 Overview

The PSSP 2.1 software provides a single point of control for administrative tasks.

9.1.1 User Accounts

SP User Management allows you to add, delete, change, and list user account information from a single point of control. Users can have the same account, home directory, and environment across all the nodes in the system.

9.1.2 File Collections

Groups of files can be defined as a file collection and changes to any files in that collection can be propagated to all nodes. The user.admin file collection is provided for propagating user administration files to the nodes on the system.

9.1.3 Auto Mount Daemon (Amd)

Amd allows filesystems to be mounted on demand when they are first referenced and then be umounted after a specific period of inactivity. SP User Management uses Amd to automount a user′s home directory when the user performs a login on a node.

9.1.4 Print Services

SP User Management provides the ability to shift printing functions from the nodes to a print server, eliminating the need to maintain and support print queues on a large number of nodes.

9.1.5 Network Time Protocol (NTP)

The RS/6000 SP system uses NTP to synchronize the time-of-day clocks on the Control Workstation and the nodes.

In the following sections of this chapter, the assumption is made that the user has chosen to implement SP User Management together with file collections and the auto mount daemon.

9.2 Components

The following options of the ssp installp image are relevant to User and Services Management. Component version numbers included are for PTF 11.

9.2.1 ssp.basic 2.1.0.10

This component includes the login control functions whereby users can either be allowed or denied access to specific nodes.

9.2.2 ssp.sysman 2.1.0.7

This component includes the following functions:

- User management
- File collections
- Auto Mount Daemon
- Print support
- Time services

9.3 Managing User Accounts

In order to have the SP User Management tools added to the System Management Interface Tool (SMIT) panels, specify “true” in the User Administration Interface field of the Site Environment Information panel of SMIT. To make use of the various services (file collections, Amd, print services, and NTP), specify “true” in the appropriate fields of the same SMIT panel.

```
                        Site Environment Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                      [Entry Fields]
  Default Network Install Image                    [bos.obj.ssp.41]
  Remove Install Image after Installs               false

  NTP Installation                                  consensus
  NTP Server Hostname(s)                           [""]
  NTP Version                                       3

  AMD Configuration                                 true

  Print Management Configuration                    false
  Print system secure mode login name              [""]

  User Administration Interface                     true
  Password File Server Hostname                    [sp21cw0]
  Password File                                    [/etc/passwd]
  Home Directory Server Hostname                   [sp21cw0]
  Home Directory Path                              [/home/sp21cw0]

  File Collection Management                        true
  File Collection daemon uid                       [102]
  File Collection daemon port                      [8431]

  SP Accounting Enabled                             false
  SP Accounting Active Node Threshold              [80]
  SP Exclusive Use Accounting Enabled               false
  Accounting Master                                [0]
```

Figure 83. Default Values for the Site Environment Information Panel

These configuration values are stored in the System Data Repository and are used as default values when managing SP user accounts with the spmkuser, spchuser, and sprmuser commands. Use the splstdata -e command to list the current values of the SP object attributes from the System Data Repository. See Figure 84 on page 216 for an example of the output from the command.

```
root@sp21cw0 / > splstdata -e ] pg

             List Site Environment Database Information

attribute                value
-------------------------------------------------------------------------
control_workstation      sp21cw0
cw_ipaddrs               9.12.0.137:9.12.60.98:9.12.60.99:
install_image            bos.obj.ssp.41
remove_image             false
primary_node             1
ntp_config               consensus
ntp_server               ""
ntp_version              3
amd_config               true
print_config             false
print_id                 ""
usermgmt_config          true
passwd_file              /etc/passwd
passwd_file_loc          sp21cw0
homedir_server           sp21cw0
homedir_path             /home/sp21cw0
filecoll_config          true
supman_uid               102
supfilesrv_port          8431
spacct_enable            false
spacct_actnode_thresh    80
spacct_excluse_enable    false
acct_master              0
cw_has_usr_clients       false
code_version             PSSP-2.1
layout_dir               /spdata/sys1/syspar_configs/1nsb0isb/config.16/l
authent_server           ssp
backup_cw                ""
ipaddrs_bucw             ""
active_cw                ""
```

Figure 84. Output from splstdata -e

The locations (server hostname and directory) for user administration files are also stored in the System Data Repository. These default values may be changed. However, the /etc/passwd and /etc/security/passwd files are updated from the Control Workstation. Therefore, if the master copy of these two files is not located on the Control Workstation, the files must be copied to the Control Workstation every time they are changed, so that the nodes will be updated with the correct password information.

9.3.1 Adding an SP User

Before adding an SP user, ensure that the host which is serving user directories (the Control Workstation in our case) has its /etc/exports file modified to grant access to all hosts that need to mount the user's home directory. For example, add /home/sp21cw0 to /etc/exports on the Control Workstation and then run exportfs -a. If this is not done, spmkuser will return the following message when adding a user (harry) on the Control Workstation (sp21cw0).

```
spmkuser: 0027-122 Warning: User directory, /home/sp21cw0/harry, is in an
unexported filesystem on sp21cw0.
```

If using the SMIT panels to add a user, only the user's name needs to be entered. The other fields will default to values contained in the System Data Repository and in the /usr/lpp/ssp/bin/spmkuser.default file. After the user has been successfully added, use the Change/Show Characteristics of a User SMIT panel to check the user's attributes.

The spmkuser command generates a random password for the user and stores it in /usr/lpp/ssp/config/admin/newpass.log. The root user has read and write permission to this file. It is the administrator's responsibility to communicate this password to the new user and to delete the contents of this file periodically.

Note: Users can only change their passwords on the Control Workstation.

The administrator must still perform the two following tasks on the Control Workstation before the new user can successfully login.

1. /etc/amd/refresh_amd
2. dsh -a /var/sysman/supper update user.admin

See the topic 9.5, "Managing Amd" on page 221 for further information on *Amd*. See the topic 9.4, "Managing File Collections" on page 218 for further information on the *File Collection*. Step 1 will refresh the *Amd* mount maps. If this command is not run before the new user attempts to login, the following error message will be seen at login time:

```
3004-614 Unable to change directory to "/u/xyz".
        You are in "/home/guest" instead.
```

Step 2 will make the new user known on all the nodes. This step avoids having to wait until all the nodes are updated automatically by supper (by default this happens at 10 past the hour every hour).

9.3.2 Changing User Attributes

When using SP User Management (usermgmt_config = true), the following base AIX commands are linked to SP commands on each node:

chfn -> /usr/lpp/ssp/config/sp_chfn

chsh -> /usr/lpp/ssp/config/sp_chsh

passwd -> /usr/lpp/ssp/config/sp_passwd

The original AIX files have been saved in their original location (/bin) with an extension .aix added. The linked *sp_* commands inform users which host to log in to when changing their password, shell, or gecos information.

```
user1@sp21n09 / > chsh
Please change your shell info (chsh) at sp21cw0.
After you have done this, it will take up to one hour for the
new shell info to propagate to all nodes.
```

Figure 85. Example of chsh Command on a Node

9.3.3 Login Control

The /usr/lpp/ssp/bin/spacs_cntrl command may be used to control interactive user access on nodes. The command updates the login and rlogin attributes in the /etc/security/user file to reflect whether or not a user may login to a node.

For example, dsh -w sp21n09 /usr/lpp/ssp/bin/spacs_cntrl block user1 would result in the attributes for *user1* in /etc/security/user on sp21n09 (node 9) to be modified as follows:

```
user1:
        login = false
        rlogin = false
        ttys = ALL,!RSH,!REXEC
```

Figure 86. Extract from /etc/security/user File

If user1 attempts to login to sp21n09, the following message will appear.

```
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1994.
login: user1
user1's Password:
Remote logins are not allowed for this account.

login:
```

Figure 87. Example of Login by Blocked User ID

The command, spacs_cntrl unblock user1, can be used to regain login access for user1 on a node.

For further information on User Access Problems, see Chapters 12 and 27 in the *IBM RISC System/6000 Scalable POWERparallel Systems: Diagnosis and Messages Guide*, GC23-3899.

9.4 Managing File Collections

A file collection is a set of files and directories that are duplicated on multiple systems in a network and are managed by tools that simplify their control and maintenance. The /var/sysman/supper command uses the Software Update Protocol (SUP) to manage RS/6000 SP file collections and transfer them across the system.

For further information on file collections and the supper command see chapter 14, “Managing File Collections,” in the *IBM RISC System/6000 Scalable POWERparallel Systems: Administration Guide*, GC23-3897. *RS/6000 SP System Management: Easy, Lean and Mean*, GG24-2563 is also a good source of information.

9.4.1 Using File Collections

If *filec = true* (use `splstdata -e` to confirm) then the following sequence will start the supfilesrv daemon on the Control Workstation.

1. /etc/rc.sp is started from /etc/inittab
2. /usr/lpp/ssp/install/bin/services_config is called from /etc/rc.sp
3. /usr/lpp/ssp/install/bin/file_config is called from services_config
4. Finally, `startsrc -s supfilesrv` is executed from filec_config

Note: Older versions of *filec_config* added an entry for *supfilesrv* to /etc/inittab. To allow support for multiple Control Workstations, the supfilesrv daemon is now under System Resource Controller (SRC) control.

```
root@sp21cw0 / > lssrc -s supfilesrv
Subsystem         Group            PID     Status
 supfilesrv                        14734   active
```

Figure 88. Output from lssrc -s supfilesrv Command

9.4.2 supper Hints and Tips

1. If you have problems with the supper command, check the log files in the /var/adm/SPlogs/filec directory. Alternatively, rerun the command using the -v option.
2. Error messages similar to those shown in Figure 89 are most likely a result of mismatched levels of PSSP code on the Control Workstation and the nodes. Ensure that PSSP components are at the same level on all nodes and the Control Workstation.

```
root@sp21cw0 / > supper update user.admin

Updating collection user.admin from server sp21cw0
supper: 0018-258 Update of collection user.admin failed.
supper: 0018-276 Login failed.
```

Figure 89. supper Error Messages When PSSP Code Is Mismatched

3. Use `lssrc -s supfilesrv` to check that the supfilesrv daemon is active on the Control Workstation and on the boot/install servers. This daemon responds to requests from the nodes to update collections of files.
4. Remember that supper is a *pull* operation. That is, the command is executed on the nodes (not on the Control Workstation) and the files are *pulled* from the Control Workstation to the nodes.
5. File collections require a unique, unused user ID: *supman*. If this ID does not exist, then supper will fail with the same error messages as those shown in Figure 89. Use the log files in /var/adm/SPlogs/filec on the nodes to obtain more detailed error messages.

```
SUP 7.24 (4.3 BSD) for file /tmp/.sf10668 at May 13 12:02:12
SUP Upgrade of user.admin at Mon May 13 12:02:12 1996
SUP Fileserver 7.12 (4.3 BSD) 27302 on sp21cw0
SUP Locked collection user.admin for exclusive access
SUP: Reason:  Unknown user supman
SUP: Improper login to supman account
SUP: Upgrade of user.admin aborted at Mon May 13 12:02:12 1996
SUP: Aborted
```

Figure 90. Extract from /var/adm/SPlogs/filec/sup5.13.96.12.02r Log File

6. If a supperupdate <file collection> command seems to execute without any errors, but the relevant files are not updated on the nodes, check the following:
   - The *supman* ID must have read access to all the files in a file collection.
   - Check that the /var/sysman/sup/<file collection>/scan file is correct and up-to-date. The *scan* file should contain the same file names as those in the /var/sysman/sup/<file collection>/list file.
   - Ensure that the date and time on all nodes and the Control Workstation are correct.
   - Make sure that the files are not listed in the /var/sysman/sup/<file collection>/refuse file. This file lists all the files which were excluded during the update process. The /var/sysman/sup/refuse file (if it exists) contains a list of files to be excluded from all the file collections.
   - Verify the contents of the /etc/ssp/server_name file. This tells *supper* from which host it should update files.
7. supper uses the Ethernet network for its communication. Ensure that TCP/IP is configured correctly.
8. Remember that boot/install servers use supperupdate to request the latest files from the Control Workstation, and that the nodes use supperupdate to request the changes from the boot/install servers. Each node executes a supperupdate every hour (at 10 minutes after the hour) from the root *crontab*. This means that a change in a master file on the Control Workstation could take 2 hours to propagate to every node (assuming that boot/install servers are being used).
9. Error messages similar to those shown in Figure 91 on page 221 are a result of a PTF packaging error in PTF 10. There are two missing files in PSSP 2.1 PTF Set 10. They are:

   ```
   /usr/lpp/ssp/filec/etc/supfilesrv
   /usr/lpp/ssp/bin/acctjob
   ```

   If you are applying PTF Set 10 on a node that has *ssp.sysman* at level 2.1.0.4 (PTF U441612) or earlier, then you must apply PTF Set 8 before installing PTF Set 10. Installing PTF Set 11, which is a full PTF, will also fix the problem. If you are applying PTF Set 10 on a node that has *ssp.sysman* at level 2.1.0.5 (PTF U441818), then you should not experience any problems.

```
root@ctrws01 / > supper -v update user.admin

Updating collection user.admin from server ctrws01
SUP 7.24 (4.3 BSD) for file /tmp/.sf30822 at MAR 24 13:01:48
SUP Upgrade of user.admin at SUN MAR 24 13:01:48 1996
SUP Locked collection user.admin for exclusive access
SUP: Improper login encryption
supper: 0018-258 Update of collection user.admin failed.
supper: 0018-276 Login failed.
SUP: Improper login to supman account
supper: 0018:258 Update of collection user.admin failed.
supper: 0018:276 Login failed.
SUP: Upgrade of user.admin aborted at SUN Mar24 13:01:48 1996.
SUP: Aborted.
```

Figure 91. supper Errors When Bypassing PTF Set 8

10. Check and confirm the correctness of the various files and directories as described in the section Understanding the File Collection Structure in the *IBM RISC System/6000 Scalable POWERparallel Systems: Administration Guide*, GC23-3897.

9.5 Managing Amd

The Berkley automounter, *Amd*, can be used to manage NFS mounting of home directories and other directories. When configured, Amd mounts filesystems on demand when they are first referenced and also unmounts them after a period of inactivity.

For further information on the use of Amd tool, see Chapter 15, “Managing Amd,” in the *IBM RS/6000 Scalable POWERparallel Systems: Administration Guide*, GC23-3897. *RS/6000 SP System Management: Easy, Lean and Mean*, GG24-2563 is also a good source of information.

9.5.1 Using Amd

To make use of Amd, set `amd_config = true` using the `spsitenv` command (use `splstdata -e` to check the value, if necessary). If `amd_config = true`, then the following sequence will start the Amd daemon at boot time.

1. /etc/rc.sp is started from /etc/inittab
2. /usr/lpp/ssp/install/bin/services_config is called from /etc/rc.sp
3. /usr/lpp/ssp/install/bin/amd_config is called from services_config
4. Finally, /etc/amd/amd_start is called from services_config
5. `amd_start` starts the Amd daemon with the parameters as shown in Figure 92.

```
/etc/amd/amd -t 16.120 -x all /var/adm/SPlogs/amd/amd.log -p /u /etc/amd/maps/amd.u
```

Figure 92. Default Options Starting the Amd Daemon

For further information on Amd options, refer to the man pages in /usr/lpp/ssp/public/amd920824upl75.tar.Z. The man pages can be extracted with the commands:

```
zcat amd920824upl75.tar.Z | tar -xvf - amd920824upl75/amd/amd.8
zcat amd920824upl75.tar.Z | tar -xvf - amd920824upl75/amq/amq.8
mv amd920824upl75/amd/amd.8 /usr/man/man8
mv amd920824upl75/amq/amq.8 /usr/man/man8
```

9.5.2 Amd Hints and Tips

1. Do not use `kill -9 <pid>` to stop the Amd daemon. Use `kill -15 <pid>` instead. Using `kill -9` can cause the Amd daemon to hang. Rebooting the affected node is the only way to recover from the hang.
2. Before running `/etc/amd/amd_start -f`, unmount all filesystems which have been mounted by Amd. This will ensure that the same filesystems are not mounted multiple times, which could potentially cause problems.

```
root@sp21n09 / > df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           16384      6960   58%     1050    26% /
/dev/hd2          589824     11936   98%    11188    16% /usr
/dev/hd9var         8192      5440   34%      334    33% /var
/dev/hd3           24576     23320    6%       40     1% /tmp
/dev/hd1            8192      7840    5%       18     2% /home
sp21cw0:/tony      40960     39312    5%        -     -  /a/sp21cw0/tony
sp21cw0:/tony      40960     39312    5%        -     -  /a/sp21cw0/tony
sp21n09:(pid3472)          0         0   -1%        -     -  /u
sp21n09:(pid3472)          0         0   -1%        -     -  /auto
sp21cw0:/tony      40960     39312    5%        -     -  /a/sp21cw0/tony
```

Figure 93. Example of Output from df Command Showing Multiple Mounts

   If multiple mounts of the same filesystem have occurred (as in Figure 93), then try to unmount all the Amd-mounted filesystems. The command `umount /a/sp21cw0/tony` will need to be run three times in the example. Then run `/etc/amd/amd_start -f`. This will clear up any problems (in most cases).

3. If filesystems which should be automounted are not, check whether the Amd daemon is running. Also make sure that the map files in /etc/amd/amd-maps are correctly configured. See Figure 93 as an example of a map file. Ensure that the map files are the same on all the nodes (use supper to update them, if necessary). If everything looks correct, restart the daemon (on the Control Workstation and each node) by using the `amd_start -f` command.

```
# This file contains the definition of the amd maps for /auto.
# /etc/amd/amd-maps/amd.auto

/defaults       type:=nfs;opts:=rw,soft;sublink:=${key}

one     host==sp21cw0;type:=link;fs:=/tony \
        host!=sp21cw0;type:=nfs;rhost:=sp21cw0;rfs:=/tony

two     host==sp21cw0;type:=link;fs:=/tony \
        host!=sp21cw0;type:=nfs;rhost:=sp21cw0;rfs:=/tony

three   host==sp21cw0;type:=link;fs:=/tony \
        host!=sp21cw0;type:=nfs;rhost:=sp21cw0;rfs:=/tony

four    host==sp21cw0;type:=link;fs:=/tony \
        host!=sp21cw0;type:=nfs;rhost:=sp21cw0;rfs:=/tony
```

Figure 94. Sample Amd Map File

The example in Figure 94 will mount the directory /tony (and its subdirectories) on the Control Workstation and on the nodes as /auto. On the Control Workstation, the original directory (/tony) will be linked to /auto.

4. After creating a new Amd map file, use dsh to copy the file to each node. Then run /etc/amd/amd_start -f to restart Amd. This makes it unnecessary to reboot all the nodes that have the new map.
5. If one of the following error messages shown in Figure 95 is returned when trying to access a filesystem mounted by Amd, ensure that the filesystem has been exported on the server.

```
root@sp21n09 / > cd /auto/one
ksh[2]: /auto/one: permission denied

root@sp21n09 / > cd /auto
root@sp21n09 /auto > cd one
ksh[2]: one:  not found
```

Figure 95. Error Messages When the Filesystem Is Not Exported

To correct this problem, add the filesystem to /etc/exports and run exportfs -a on the server.

```
root@sp21n09 / > cd auto/one
root@sp21n09 /auto/one > ls -l
total 0
-rwxrwxrwx   1 root     system         0 May  9 10:04 a.1
-rwxrwxrwx   1 root     system         0 May  9 10:04 b.1
-rwxrwxrwx   1 root     system         0 May  9 10:04 c.1
```

Figure 96. Output after Exporting /tony on the Control Workstation

6. Sometimes problems with Amd may be caused by the Network File System (NFS) subsystem. Use the command `lssrc -g nfs` to check the status of the NFS subsystem, as shown in Figure 97 on page 224.

```
root@sp21n09 / > lssrc -g nfs
Subsystem         Group            PID     Status
 biod             nfs              5836    active
 nfsd             nfs              7146    active
 rpc.mountd       nfs              8968    active
 rpc.statd        nfs              9496    active
 rpc.lockd        nfs              9768    active
```

Figure 97. Example of Output from lssrc -g nfs Command

7. Use the /var/adm/SPlogs/amd/amd.log file to get further information on Amd problems. Figure 98 shows the log entries when `amd_start -f` is run successfully.

```
May 10 13:31:06 sp21n09 amd[13924]/info:  My ip addr is 0x90c3c09
May 10 13:31:06 sp21n09 amd[10598]/info:  file server localhost type locals starts up
May 10 13:31:06 sp21n09 amd[10598]/map:   Trying mount of /etc/amd/amd-maps/amd.u
on /u fstype toplvl
May 10 13:31:06 sp21n09 amd[10598]/map:   Trying mount of /etc/amd/amd-maps/amd.auto
on /auto fstype toplvl
/etc/amd/amd_start: kill -TERM 11324 ..

/etc/amd/amd_start: Starting Amd on Fri May 10 13:31:04 CDT 1996
/etc/amd/amd_start: Started Amd on Fri May 10 13:31:04 CDT 1996
May 10 13:31:06 sp21n09 amd[10598]/warn:  NIS domain name is not set.  NIS ignored.
May 10 13:31:06 sp21n09 amd[10598]/info:  /etc/amd/amd-maps/amd.u mounted fstype toplvl
on /u
May 10 13:31:06 sp21n09 amd[10598]/info:  /etc/amd/amd-maps/amd.auto mounted fstype
toplvl on /auto
/etc/amd/amd_start: kill -TERM 11324 ..
```

Figure 98. Extract from amd.log File after amd_start -f Command

Figure 99 shows the log entries when trying to change directory (cd) to an automounted directory which has not been exported on the server.

```
May 10 13:43:57 sp21n09 amd[10598]/map:  key one: map selector host (=sp21n09) did
not match sp21cw0
May 10 13:43:57 sp21n09 amd[10598]/map:  Map entry host==sp21cw0;type:=link;fs:=/tony
for /auto/one failed to match
May 10 13:43:57 sp21n09 amd[10598]/map:  Trying mount of sp21cw0:/tony on /auto/one
fstype nfs
May 10 13:43:57 sp21n09 amd[10598]/info:  Filehandle denied for "sp21cw0:/tony"
```

Figure 99. Extract from amd.log When Filesystem Is Not Exported

9.6 Managing Print Services

Print management tools are supplied with PSSP 2.1 to shift printing functions from the nodes to a print server, eliminating the need to maintain and support print queues on a large number of nodes. An RS/6000 SP node is designed to use a unique remote host, PRINT_HOST, as a print server.

For further information on the use of the SP Print Management System see Chapter 16 “Managing Print Service” in the *IBM RISC System/6000 Scalable POWERparallel Systems: Administration Guide*, GC23-3897.

9.6.1 Using Print Services

The SP Print Management System can run in either *Open*, or *Secure* Mode.

For *Secure* Mode set *print_config = secure* and *print_id = prtid*. For *Open* Mode, set *print_config = open*. Use either `smitty site_env_dialog` or `spsitenv` to set these values.

Secure Mode requires only one user (*prtid*) to have `rsh` privileges on the print server. This user account is responsible for transferring all print jobs to the print server.

Open Mode requires all users who need access to a printer to have `rsh` privileges on the print server. Therefore, these users can execute other commands on the print server through `rsh` as well, which could pose a security exposure.

If *print_config = secure* or *open*, then the following sequence will configure Print Services:

1. /etc/rc.sp is started from /etc/inittab
2. /usr/lpp/ssp/install/bin/services_config is called from /etc/rc.sp
3. /usr/lpp/ssp/install/bin/print_config is called from services_config

The print_config script copies all the print commands in /usr/bin to *cmd.AIX*. The base AIX print commands are then linked to /usr/lpp/ssp/bin/pmswitch. If Secure Mode is used, then /usr/lpp/ssp/bin/pmbec has setuid permission added.

9.7 Managing NTP

A mechanism to synchronize clocks is needed because the RS/6000 SP nodes do not have system batteries. Network Time Protocol (NTP) is used as the default mechanism, on the RS/6000 SP system, to achieve this. The RS/6000 SP uses NTP to set the time at system initialization and to synchronize time-of-day clocks on the Control Workstation and nodes. Other time-service protocols (timed or Digital Time Service) may be used instead of NTP.

For further information on NTP, see chapter 13 “Managing NTP” in the *IBM RISC System/6000 Scalable POWERparallel Systems: Administration Guide*, GC23-3897. Chapter 5 “The Network Time Protocol” in *RS/6000 SP System Management: Easy, Lean and Mean*, GG24-2563 is also a good source of information.

9.7.1 Using NTP

If `ntp_config = [consensus/timemaster/internet]` (use `splstdata -e` to confirm), then the following sequence will start the xntp daemon on the Control Workstation and the nodes:

1. /etc/rc.sp is started from /etc/inittab
2. /usr/lpp/ssp/install/bin/services_config is called from /etc/rc.sp
3. /usr/lpp/ssp/install/bin/ntp_config is called from services_config
4. /etc/rc.ntp is called from ntp_config
5. /usr/lpp/ssp/bin/xntp is started from rc.ntp

The ntp_config script creates the /etc/ntp.conf and /etc/rc.ntp files. The ntp.conf file indicates which hosts can provide time service to which other hosts.

For further information on NTP, refer to the man pages in /usr/lpp/ssp/public/ntp.tar.Z. The man pages can be extracted with the commands:

```
zcat ntp.tar.Z | tar -xvf - ntp/doc/xntpd.8
zcat ntp.tar.Z | tar -xvf - ntp/doc/xntpdc.8
zcat ntp.tar.Z | tar -xvf - ntp/doc/ntpq.8
zcat ntp.tar.Z | tar -xvf - ntp/doc/ntpdate.8
zcat ntp.tar.Z | tar -xvf - ntp/doc/ntptrace.8
mv ntp/doc/xntp*.8 /usr/man/man8
mv ntp/doc/ntp*.8 /usr/man/man8
```

9.7.2 NTP Hints and Tips

1. If the time difference between the Kerberos authentication server and the nodes is not greater than 5 minutes, then the Kerberos-authenticated commands will continue to execute.
2. Use `kill -15 <NTP pid>` to kill the NTP daemon before restarting it.
3. Use /etc/rc.ntp to start the NTP daemon, if necessary.
4. How to change the system time.
   - Create a .rhost file. This will be needed to execute `rsh` commands after the Control Workstation time has been changed (as Kerberos commands will no longer work).

```
root@sp21cw0 / > cat /.rhosts
sp21cw0 root
root@sp21cw0 / >
```

Figure 100. Example of /.rhosts File

   - Use `pcp -a /.rhost /` to copy the .rhost file to all the nodes.
   - Use `smitty date` to change the system date on the Control Workstation.
   - Execute the following script on the Control Workstation:

```
for i in hostlist -av
do
/usr/bin/rsh $i ’kill -15 ps -e | grep xntp | grep -v grep |
cut -c1-6’
/usr/bin/rsh $i /etc/rc.ntp
done
```

This script will stop the NTP daemon on each node and then restart it. This will synchronize the time on the nodes with the Control Workstation.

- Use `dsh -a rm /.rhost` to delete the .rhost file on all the nodes. Notice that because the time is synchronized, Kerberos commands work again.

Appendix A. RS/6000 SP Script Files

Many of the SP tasks are carried out by script files. This appendix provides flow charts for the main PSSP scripts, which can be used as a reference for problem determination.

A.1 The setup_authent Script

The setup_authent script provides the initial setup for the Kerberos authentication services. This script has several built-in functions which are included in separate flow charts to keep the descriptions clear.

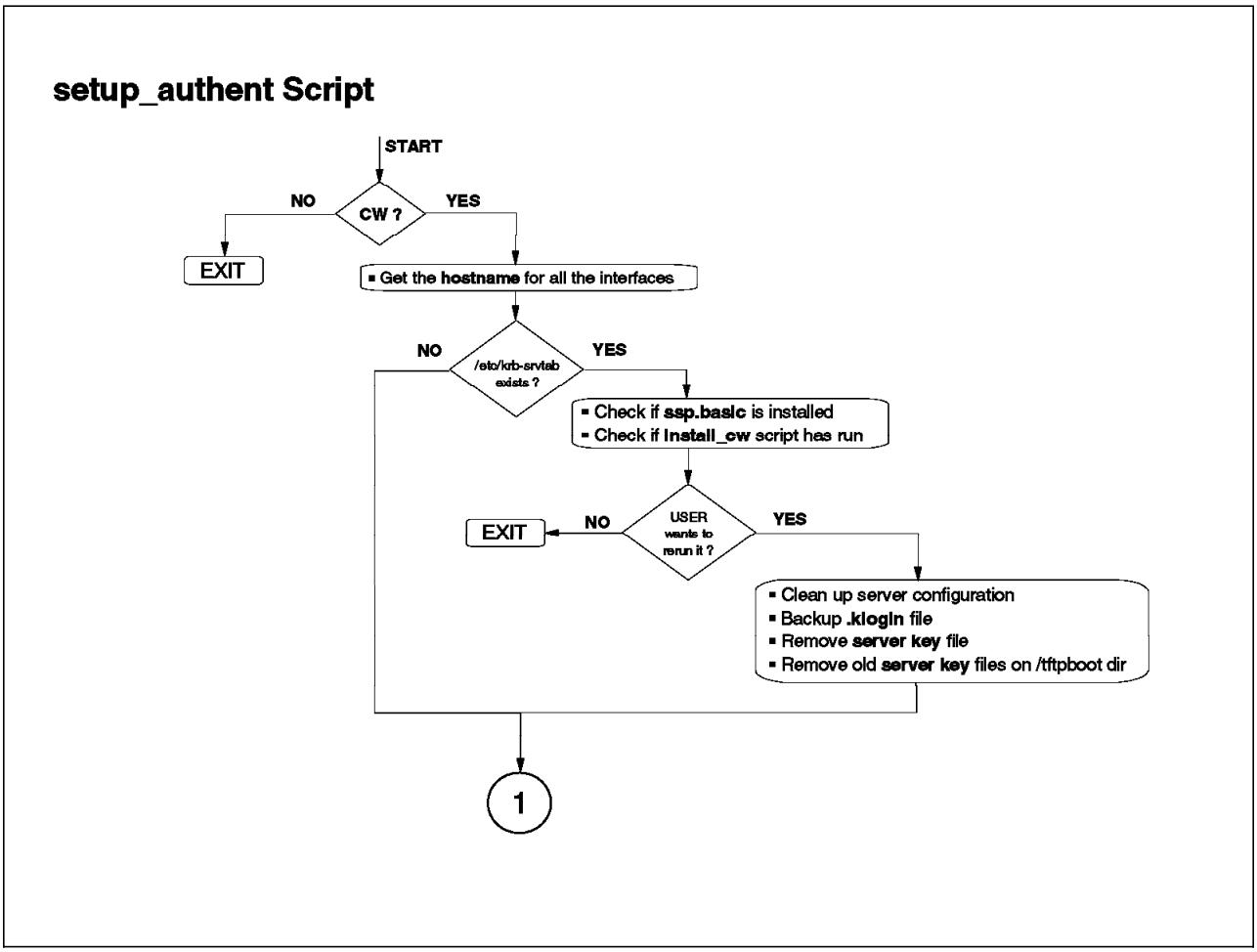


Figure 101. setup_authent Script Flow Chart (1/7)

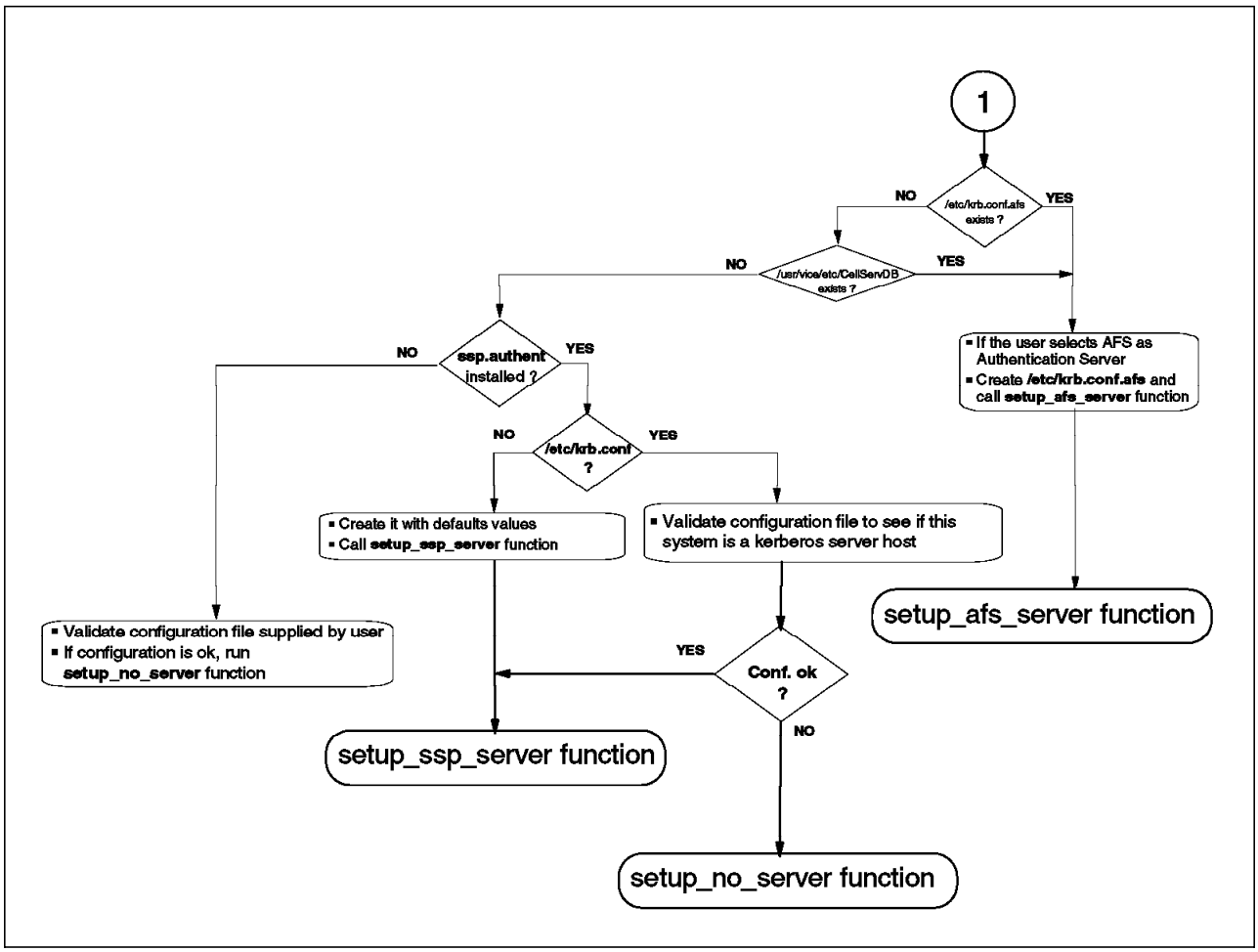


Figure 102. setup_authent Script Flow Chart (2/7)

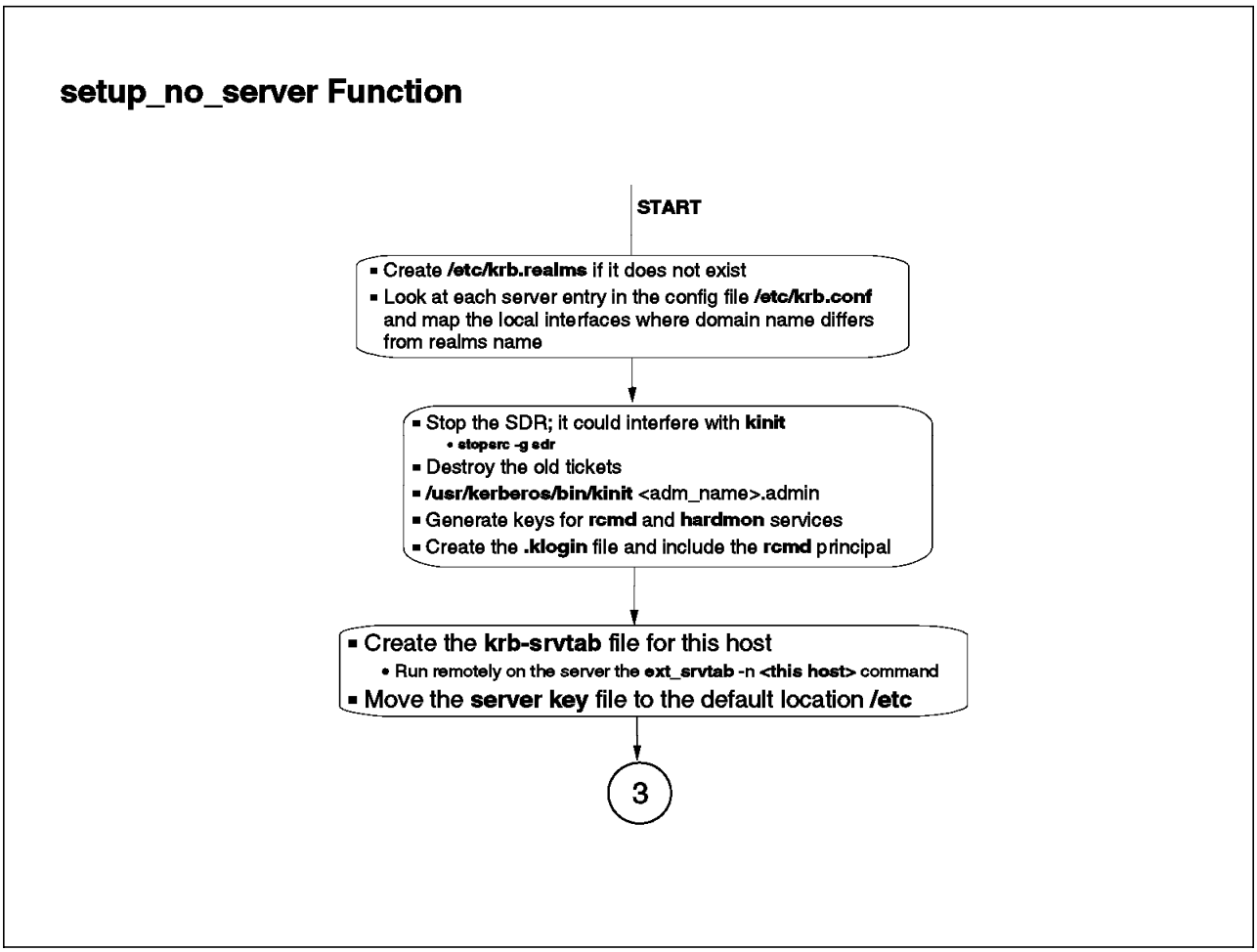


Figure 103. setup_authent Script Flow Chart (3/7)

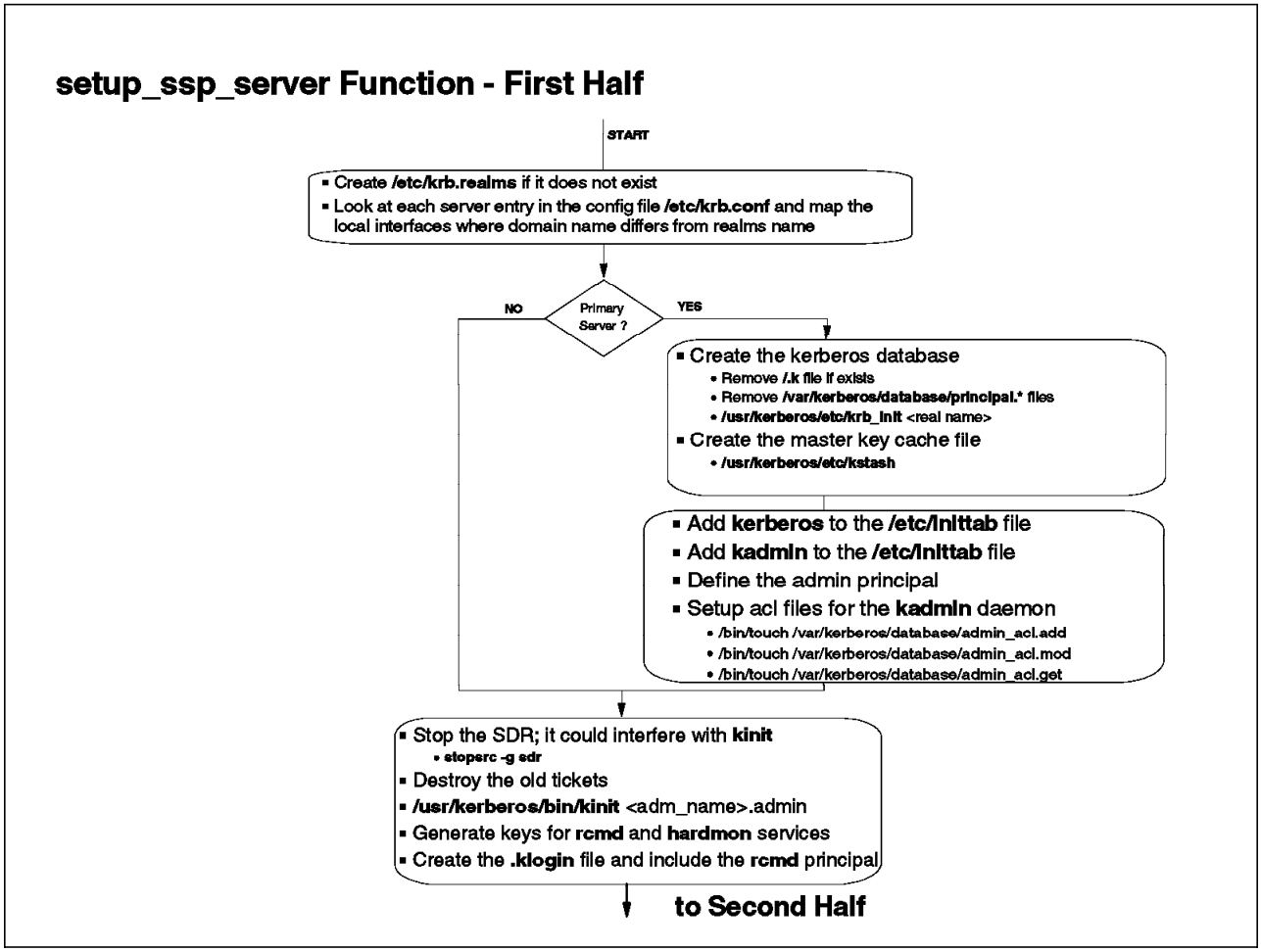


Figure 104. setup_authent Script Flow Chart (4/7)

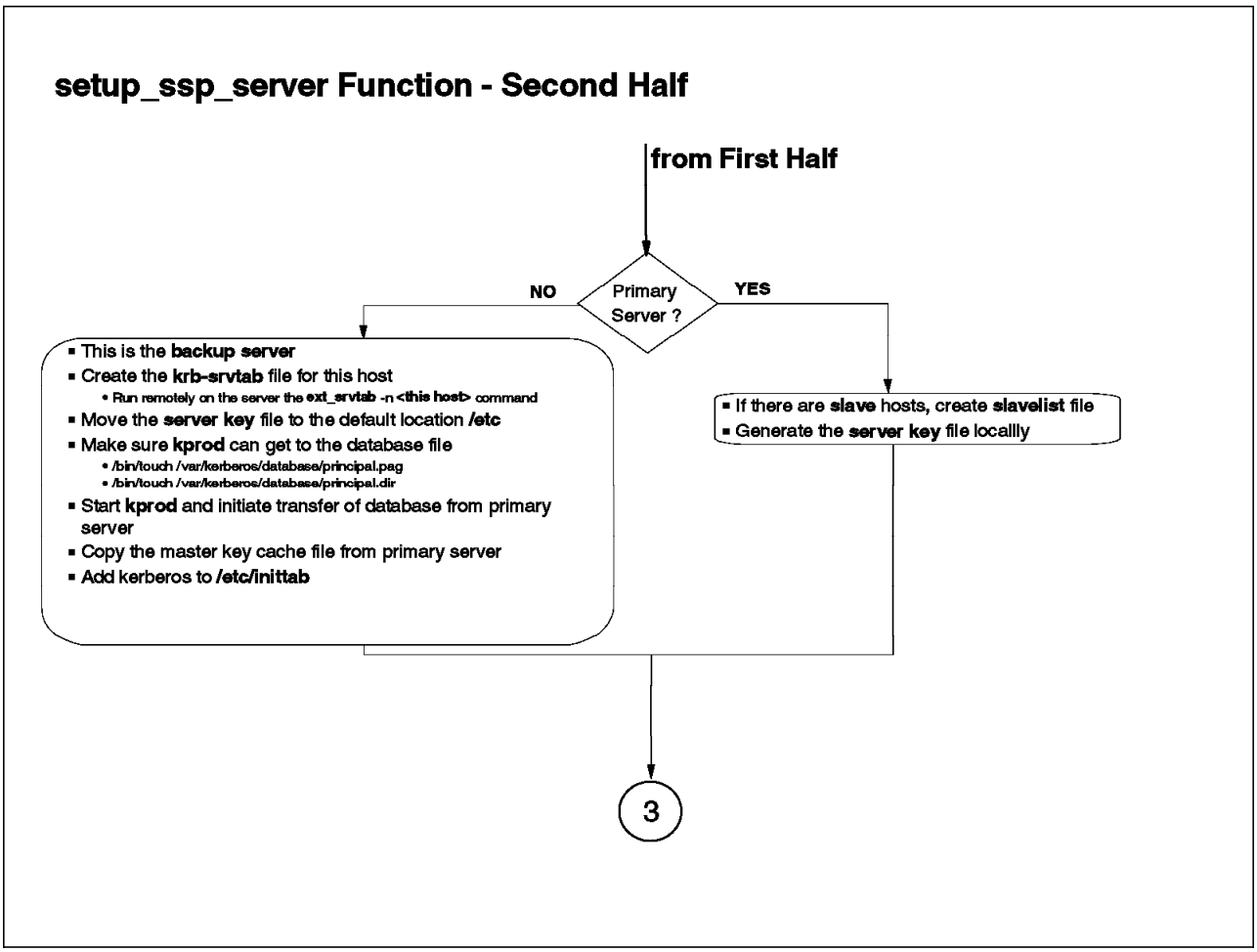


Figure 105. setup_authent Script Flow Chart (5/7)

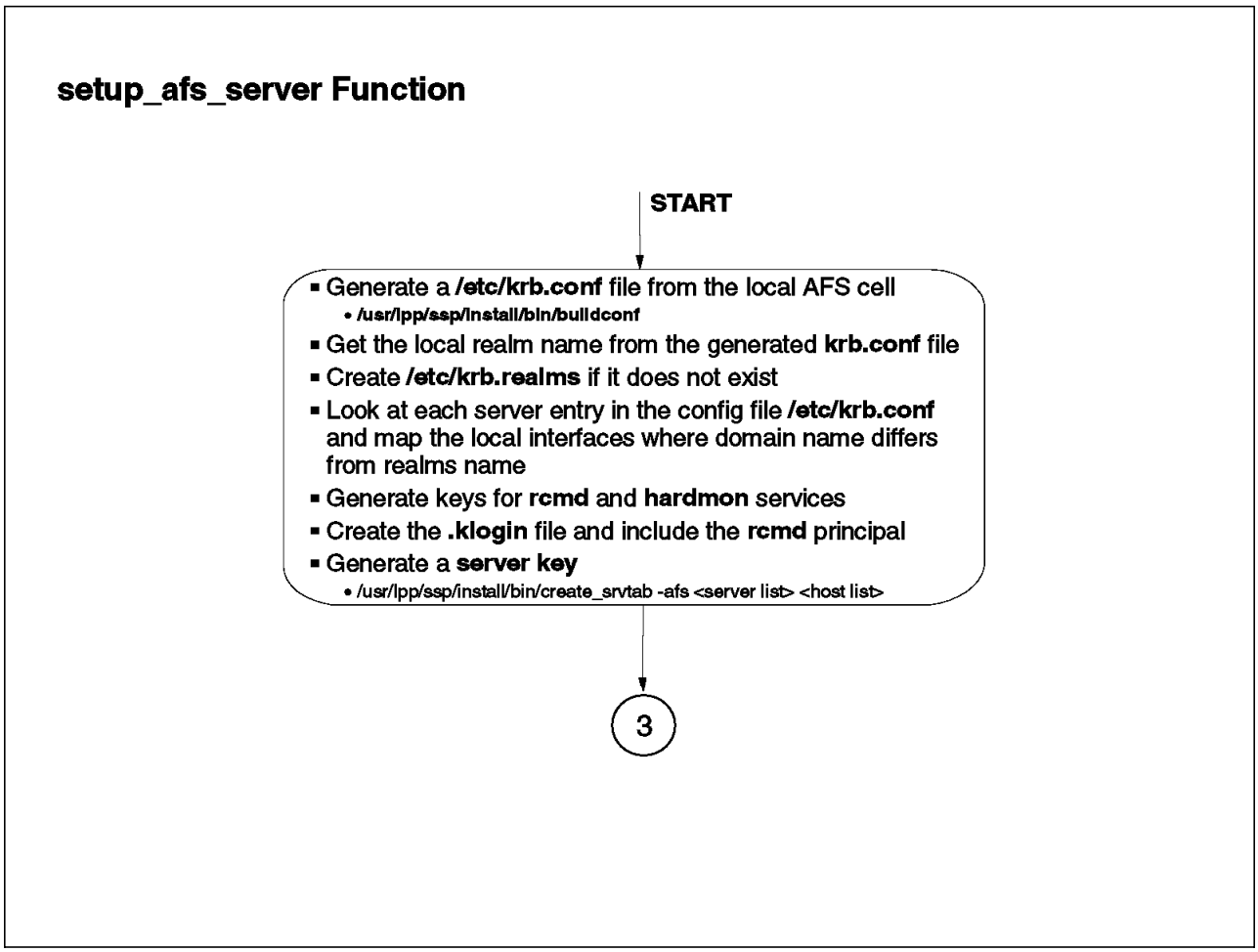


Figure 106. setup_authent Script Flow Chart (6/7)

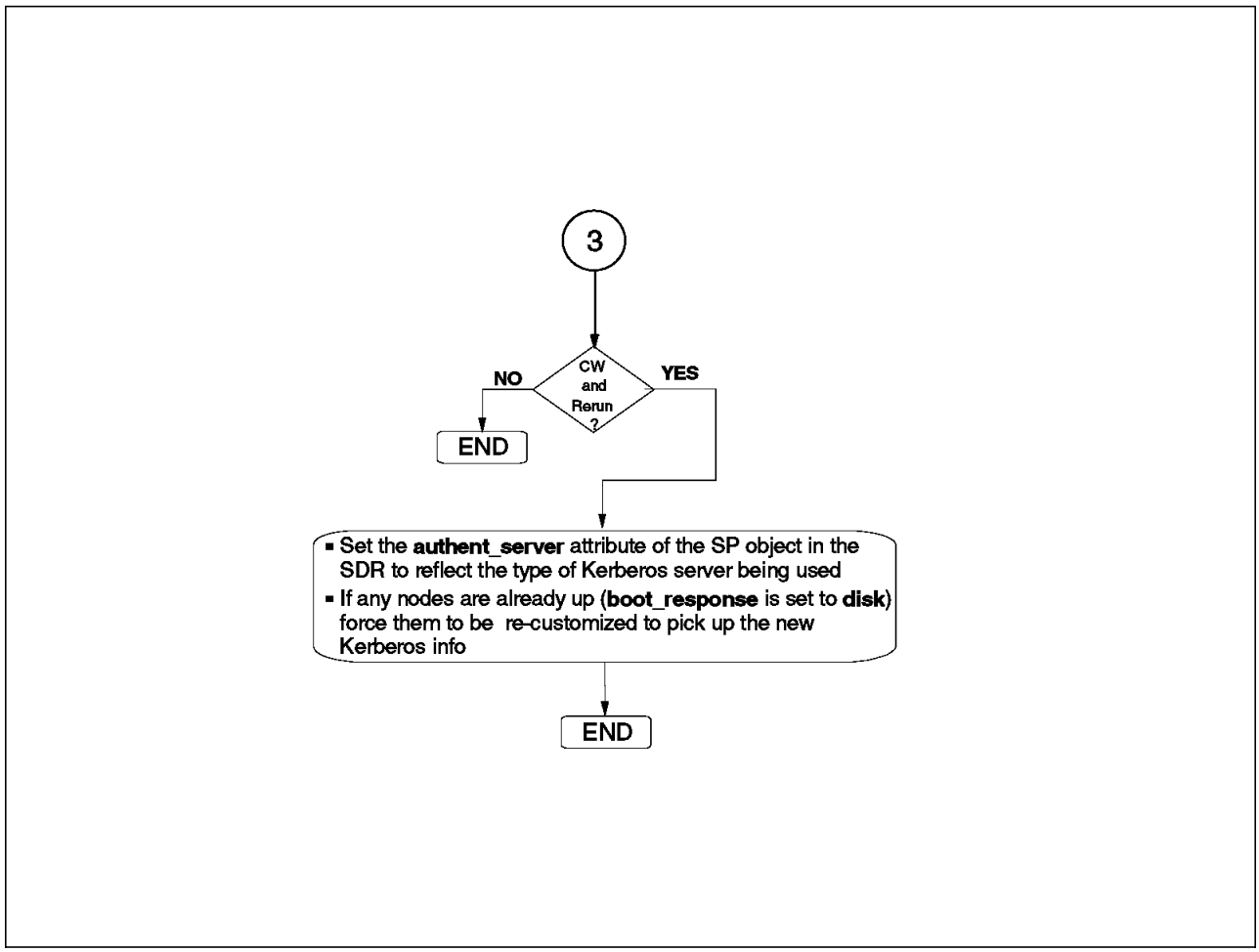


Figure 107. setup_authent Script Flow Chart (7/7)

A.2 The install_cw Script

This script is invoked during the installation procedure. The CW is installed here, creating the directory structure and setting up the process and subsystems infrastructure that will manage the RS/6000 SP.

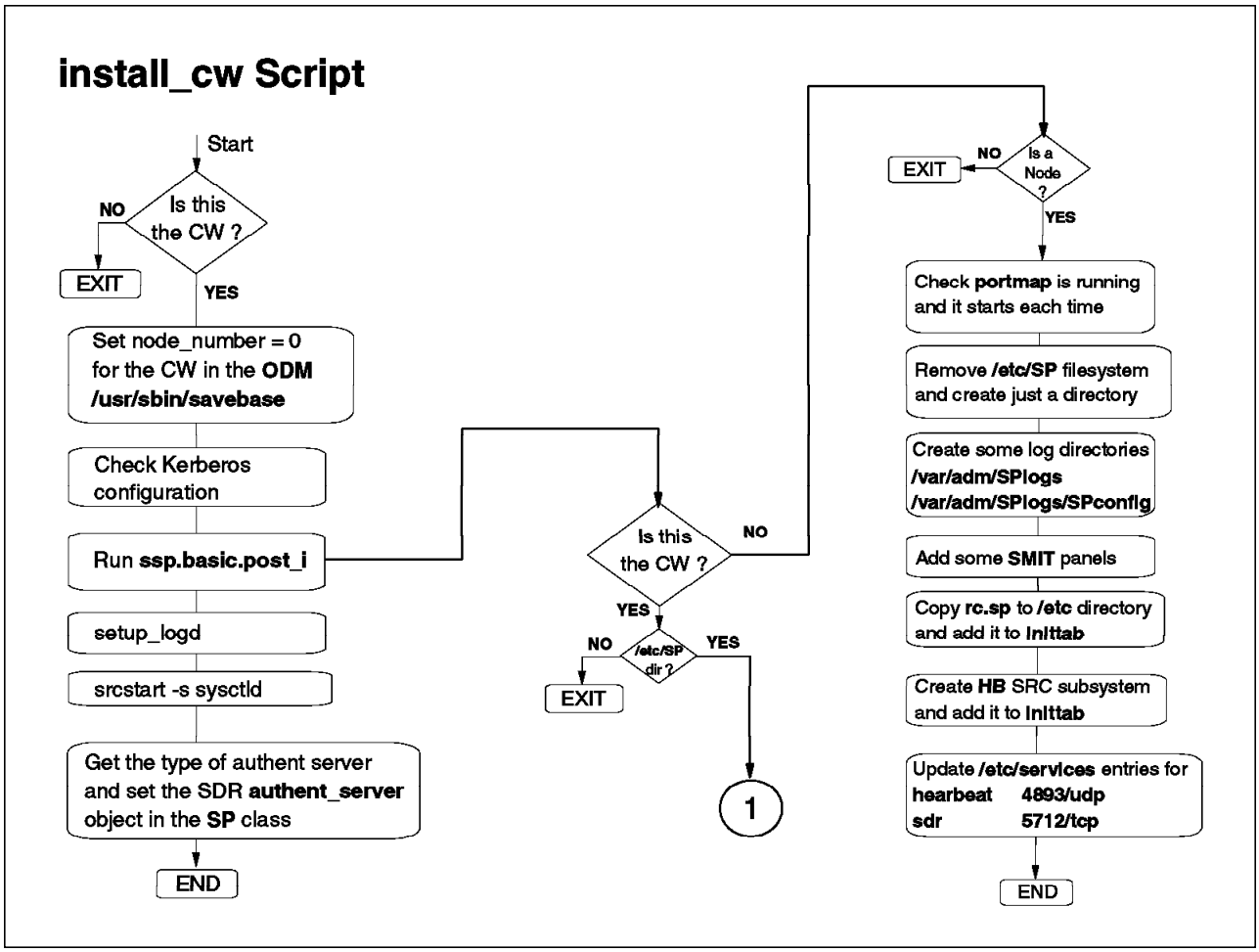


Figure 108. install_cw Script Flow Chart (1/3)

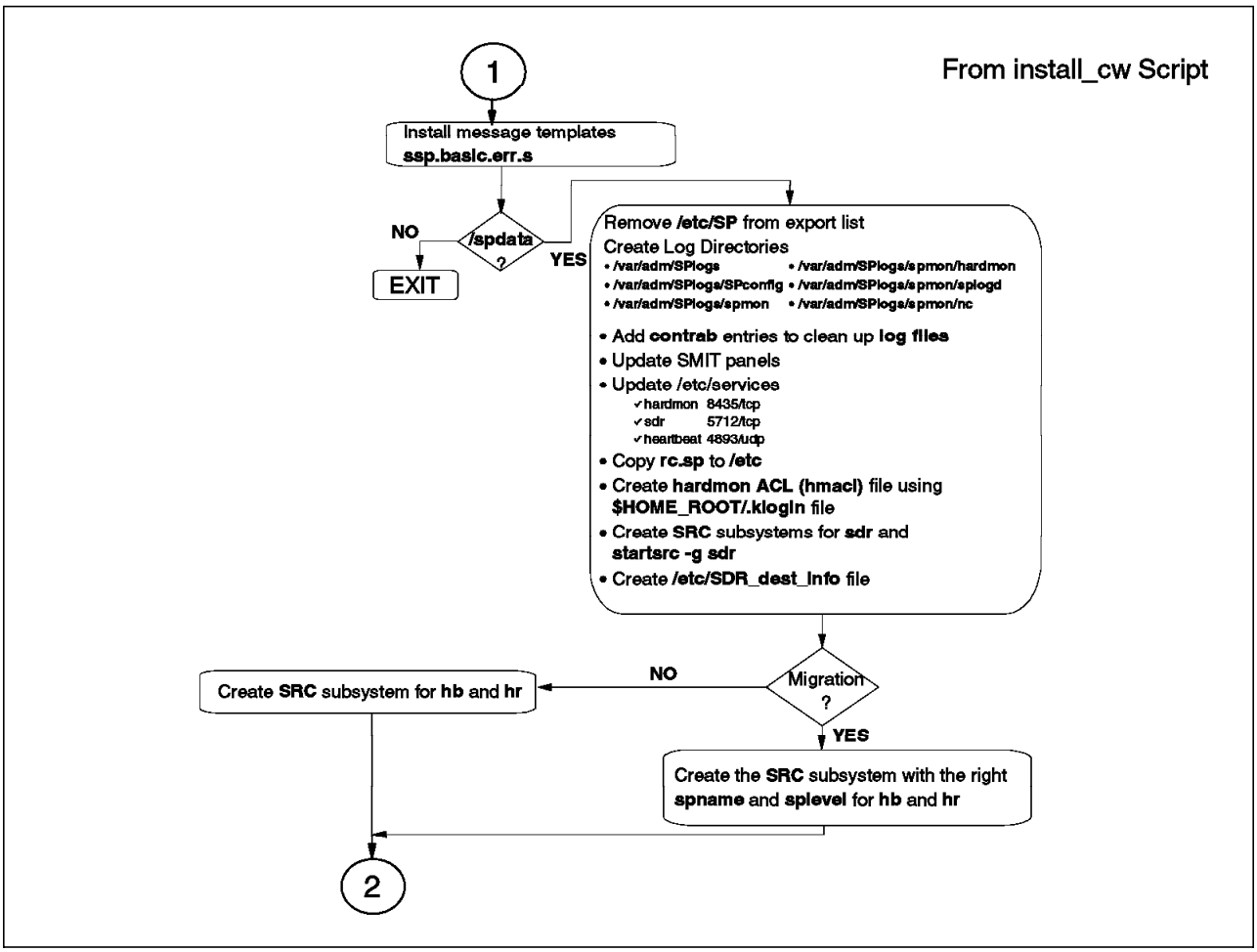


Figure 109. install_cw Script Flow Chart (2/3)

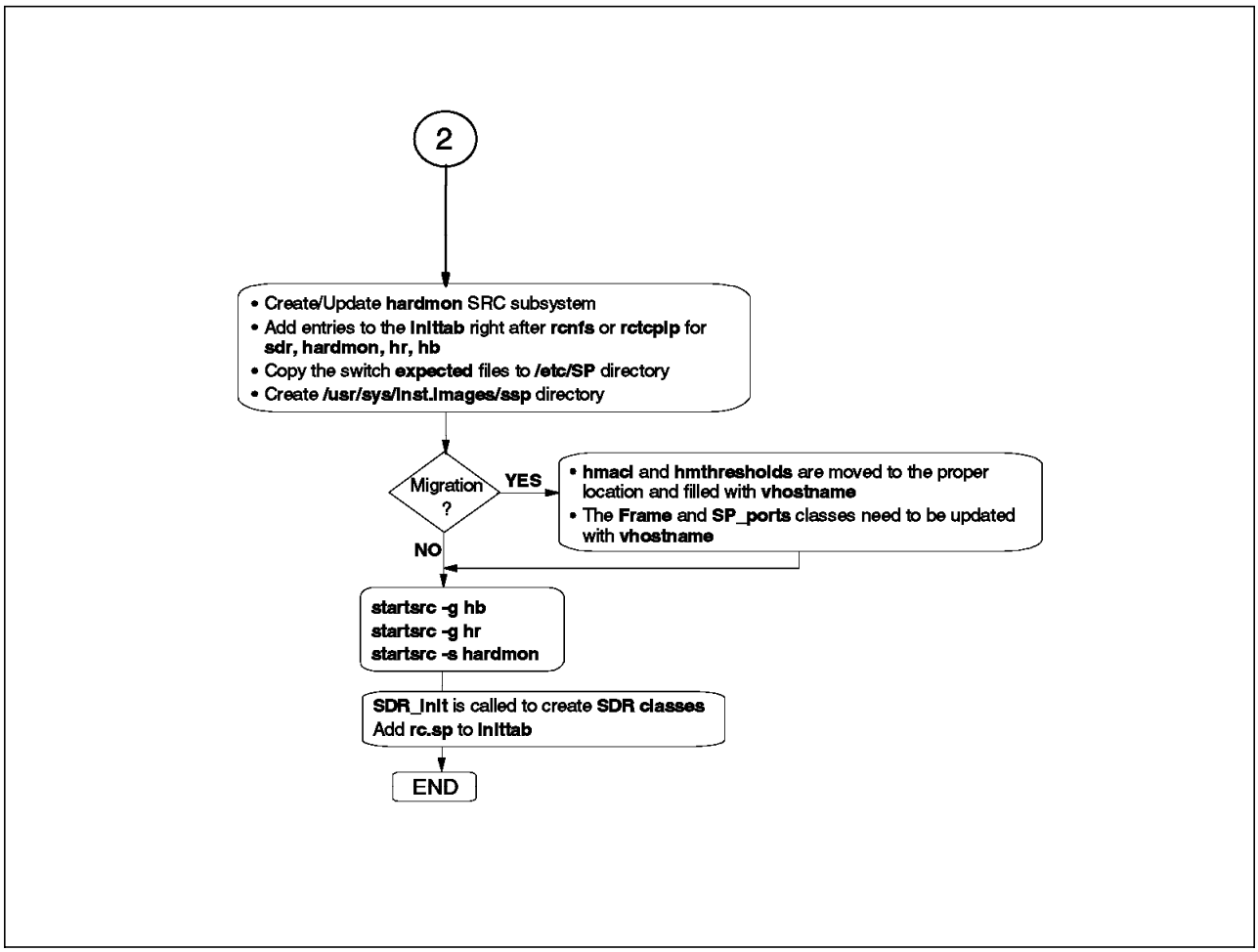


Figure 110. install_cw Script Flow Chart (3/3)

A.3 The setup_server Script

This script is run to check and configure boot/install servers. It can be run from the Control Workstation or from the nodes which will be used as boot/install servers.

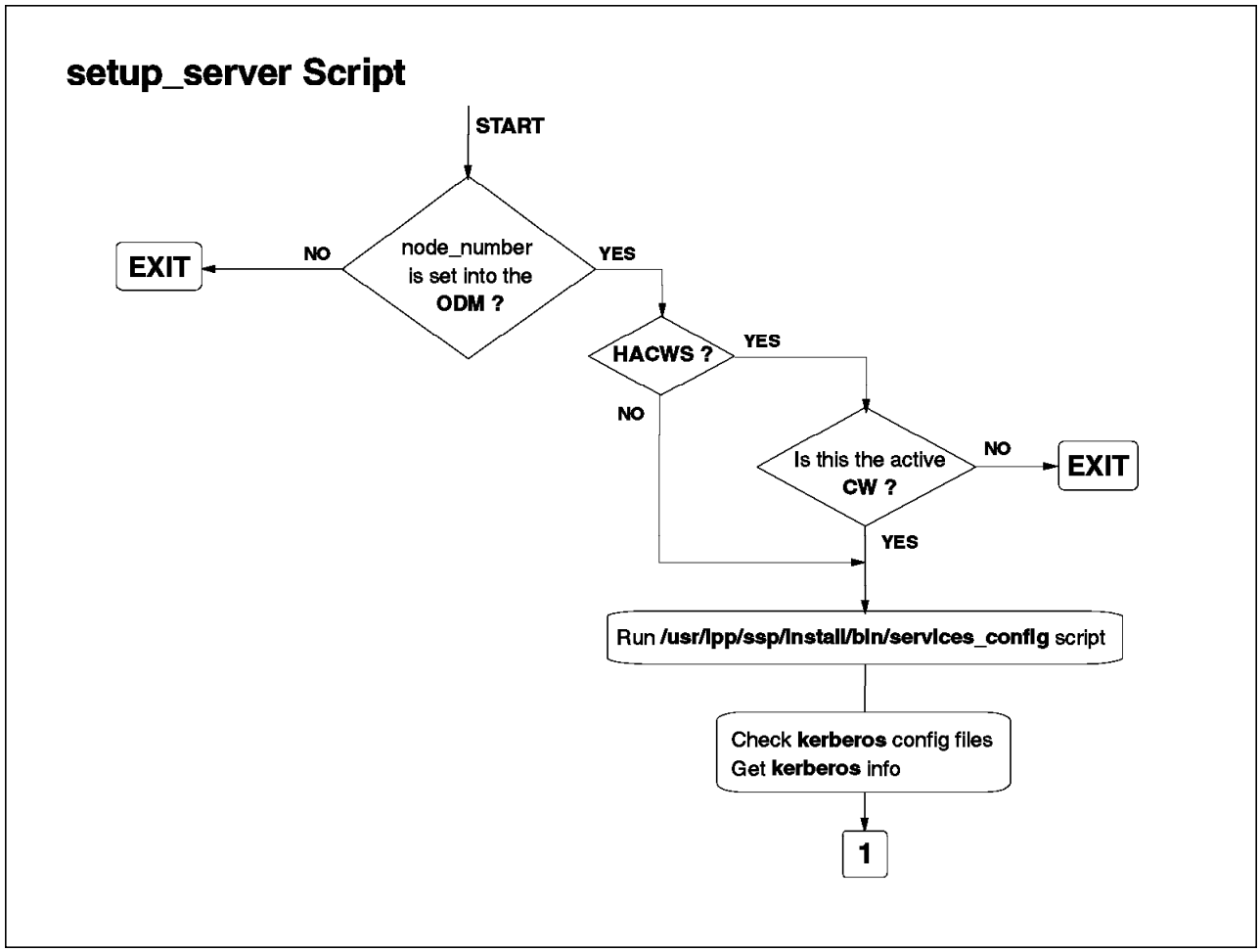


Figure 111. setup_server Script Flow Chart (1/23)

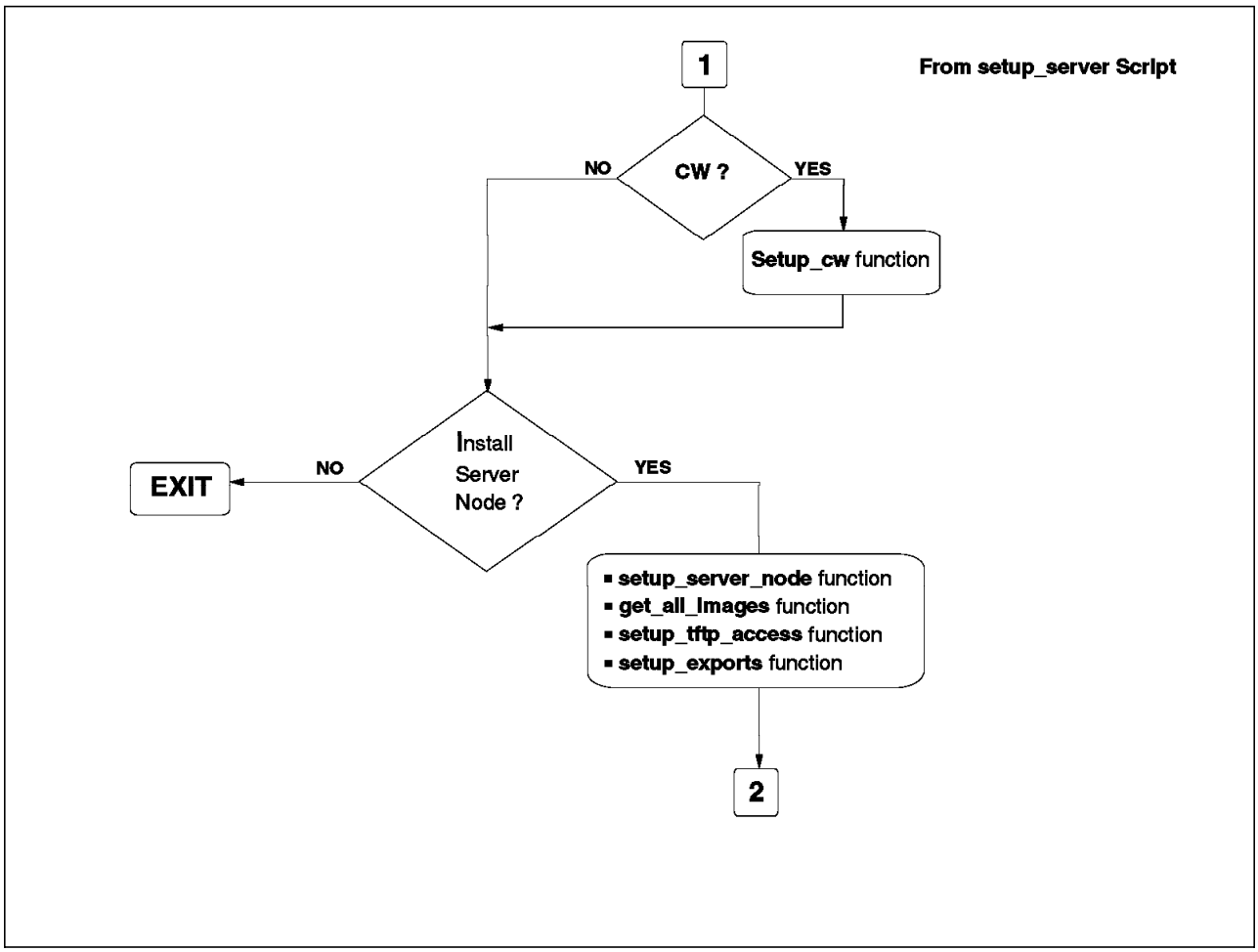


Figure 112. setup_server Script Flow Chart (2/23)

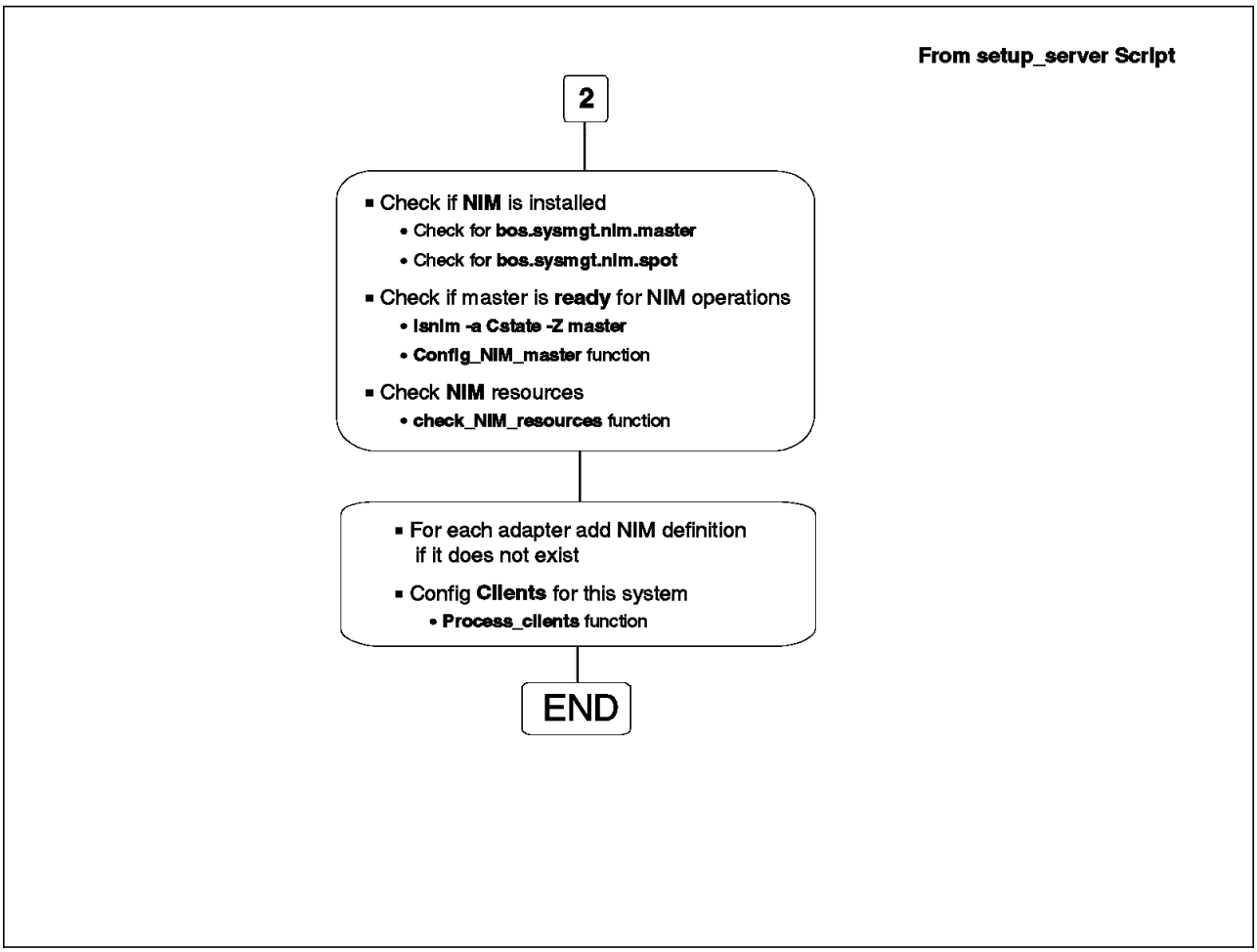


Figure 113. setup_server Script Flow Chart (3/23)

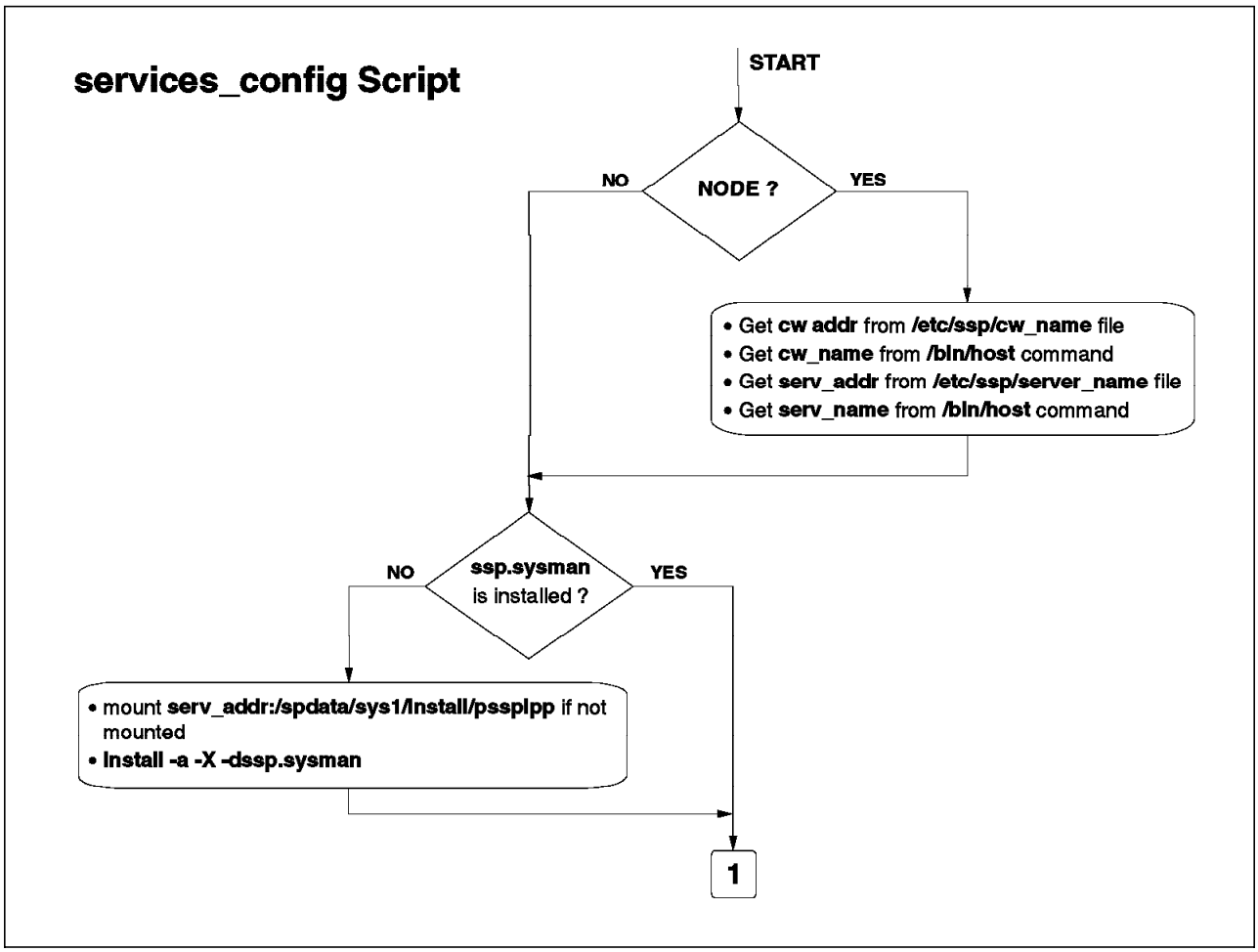


Figure 114. setup_server Script Flow Chart (4/23)

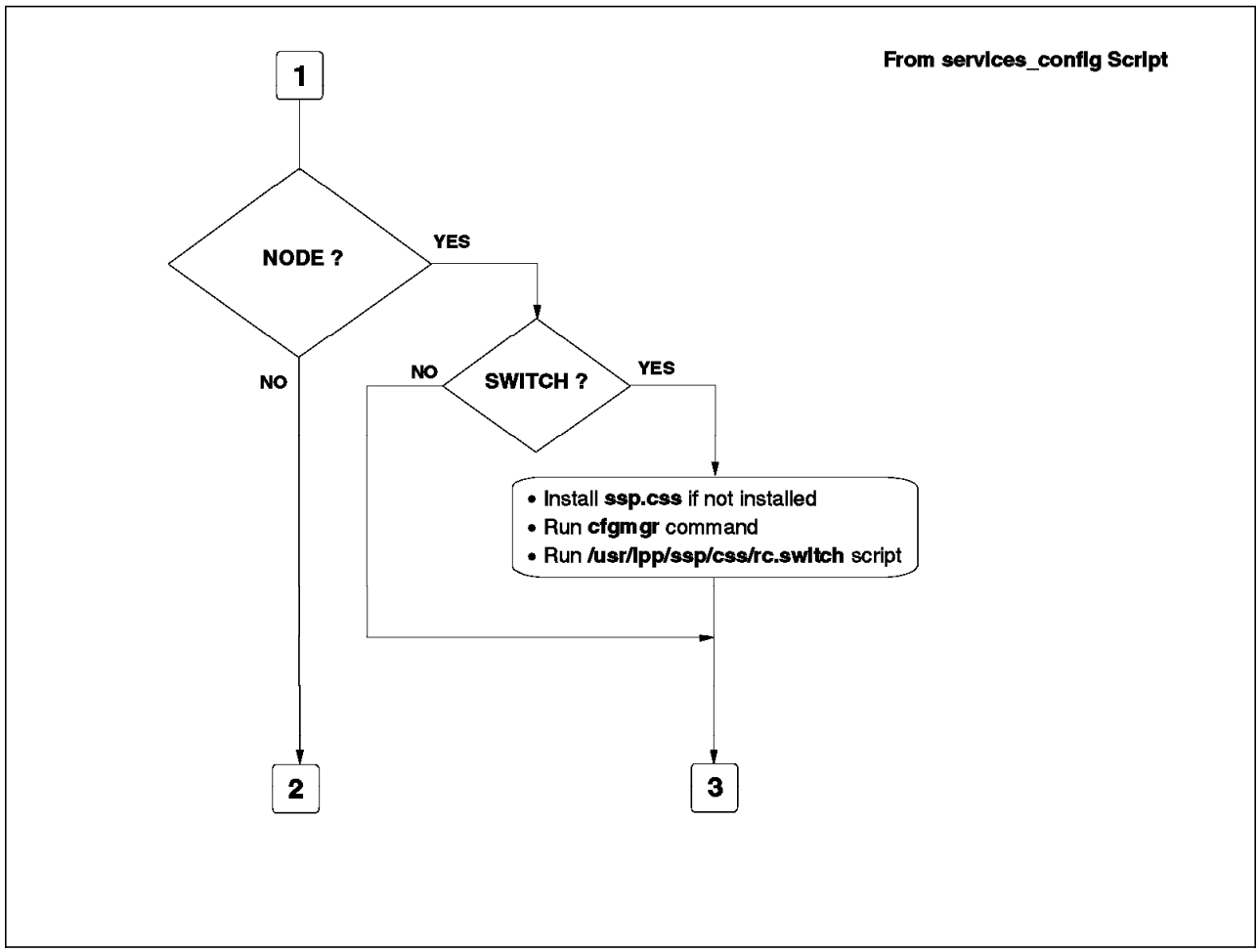


Figure 115. setup_server Script Flow Chart (5/23)

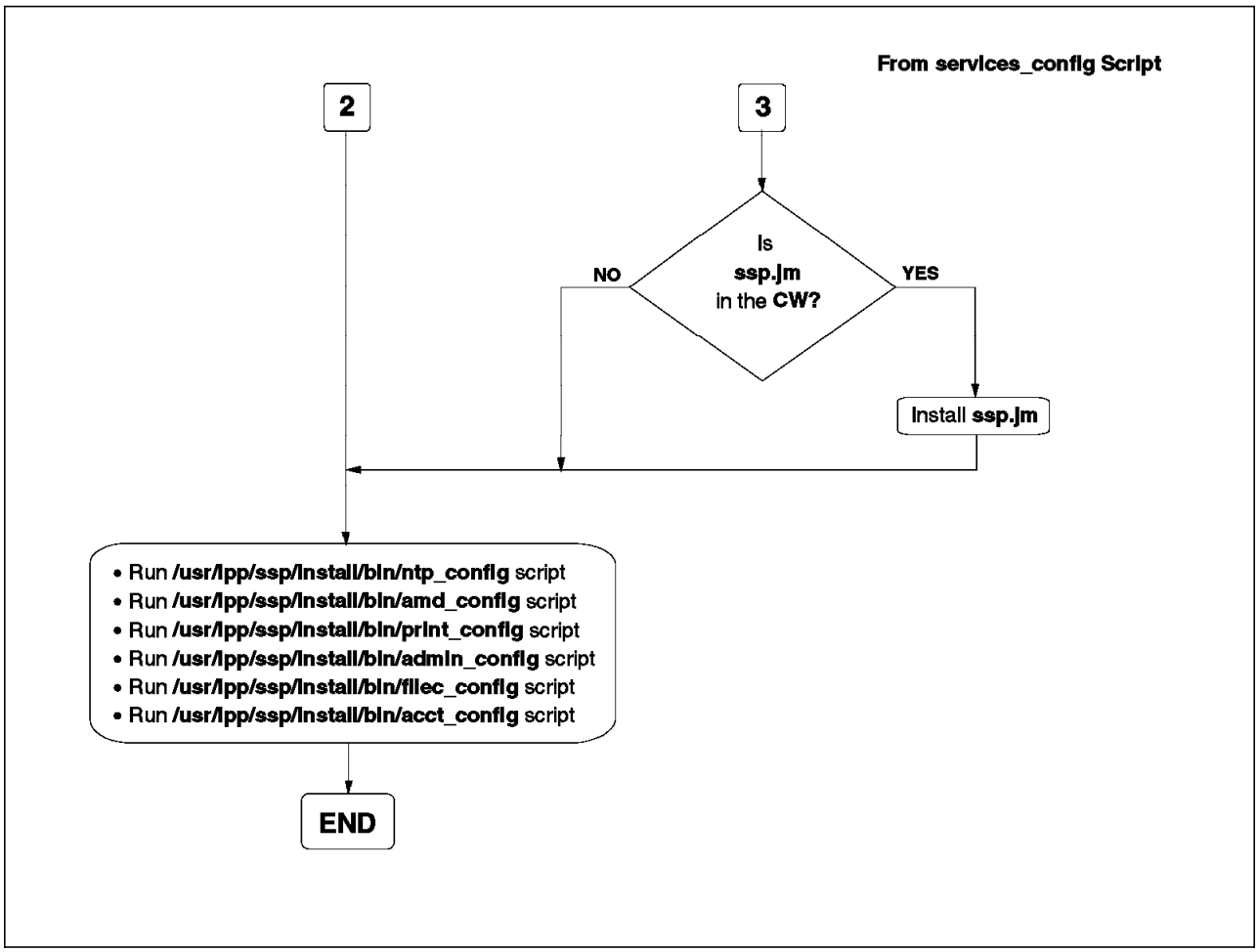


Figure 116. setup_server Script Flow Chart (6/23)

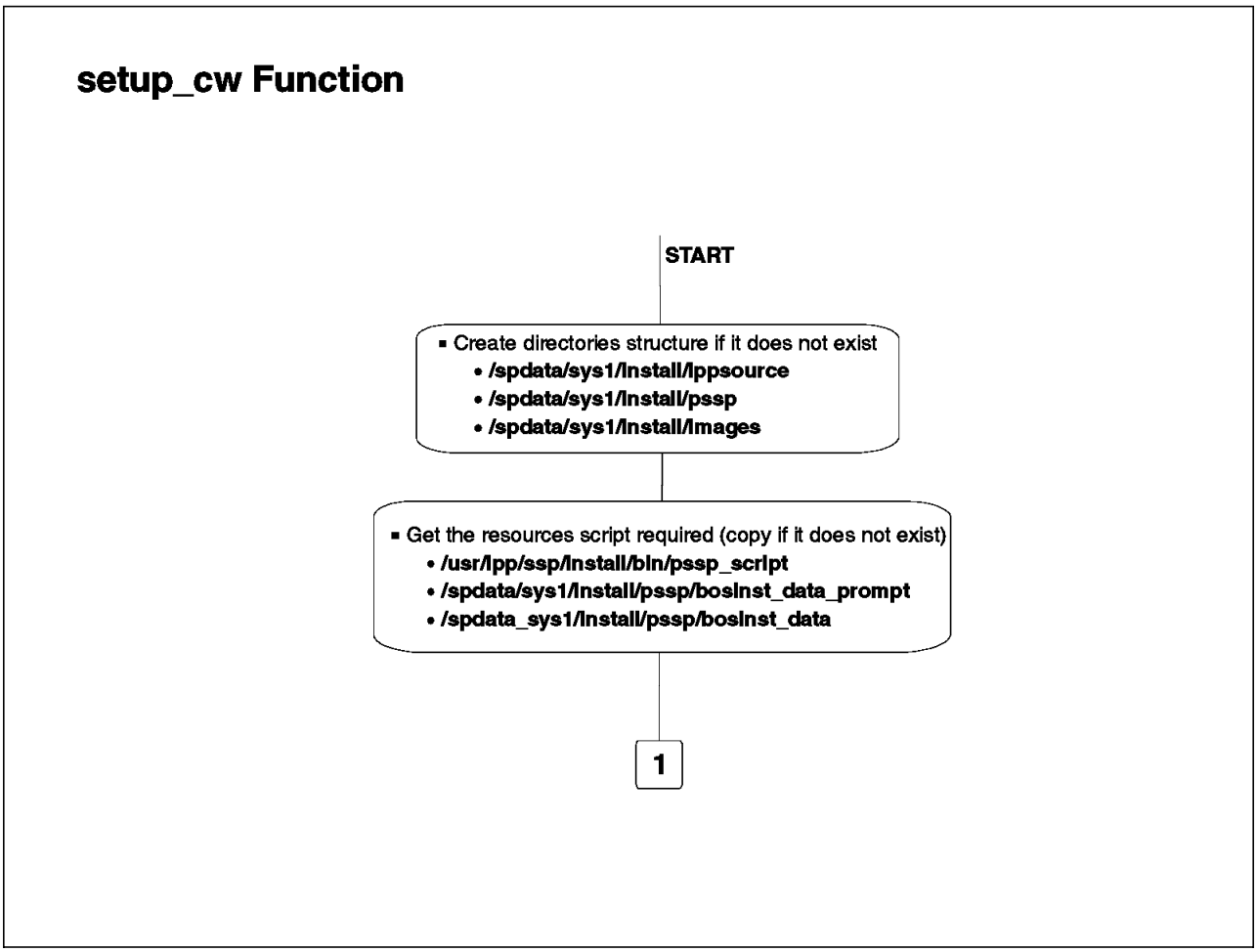


Figure 117. setup_server Script Flow Chart (7/23)

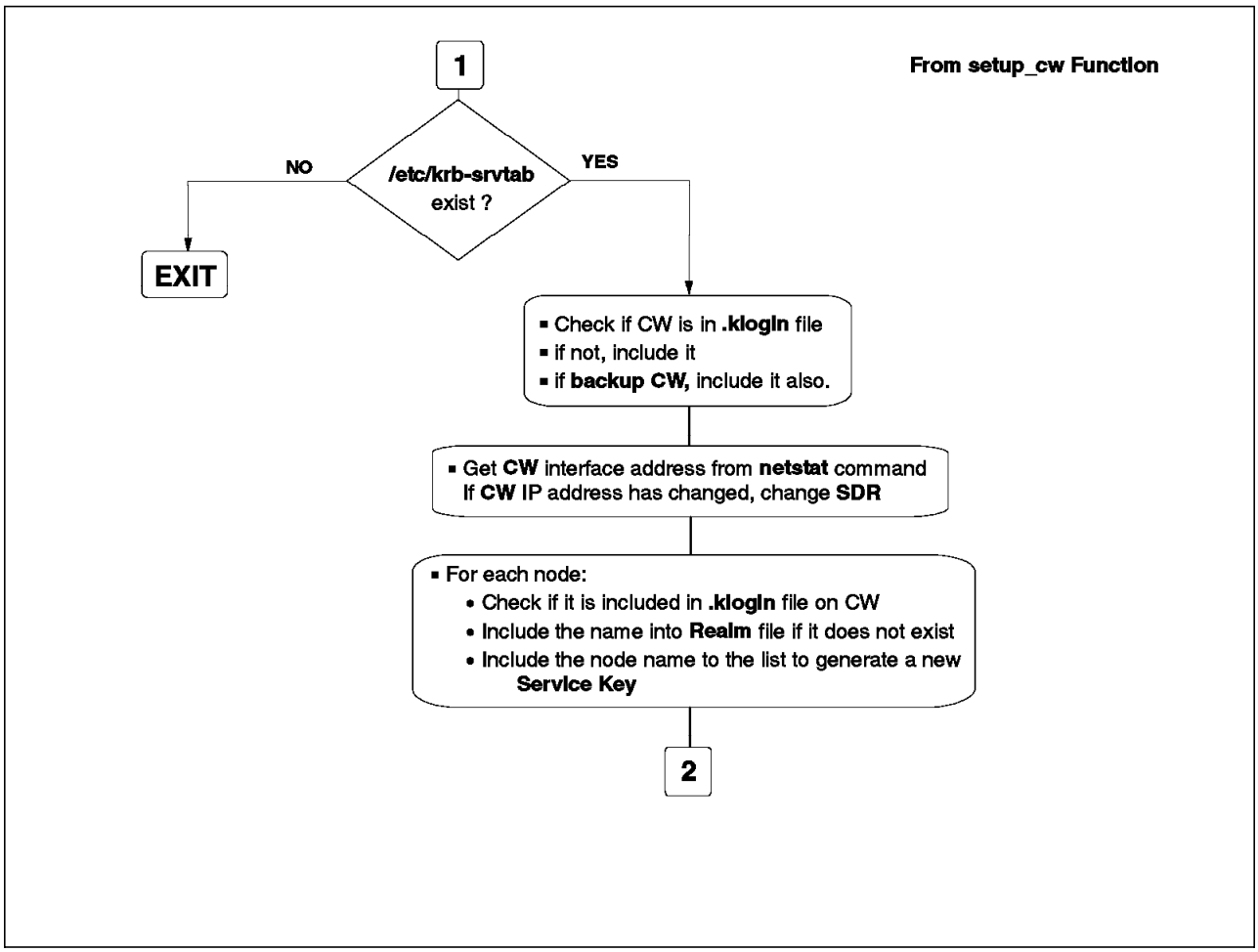


Figure 118. setup_server Script Flow Chart (8/23)

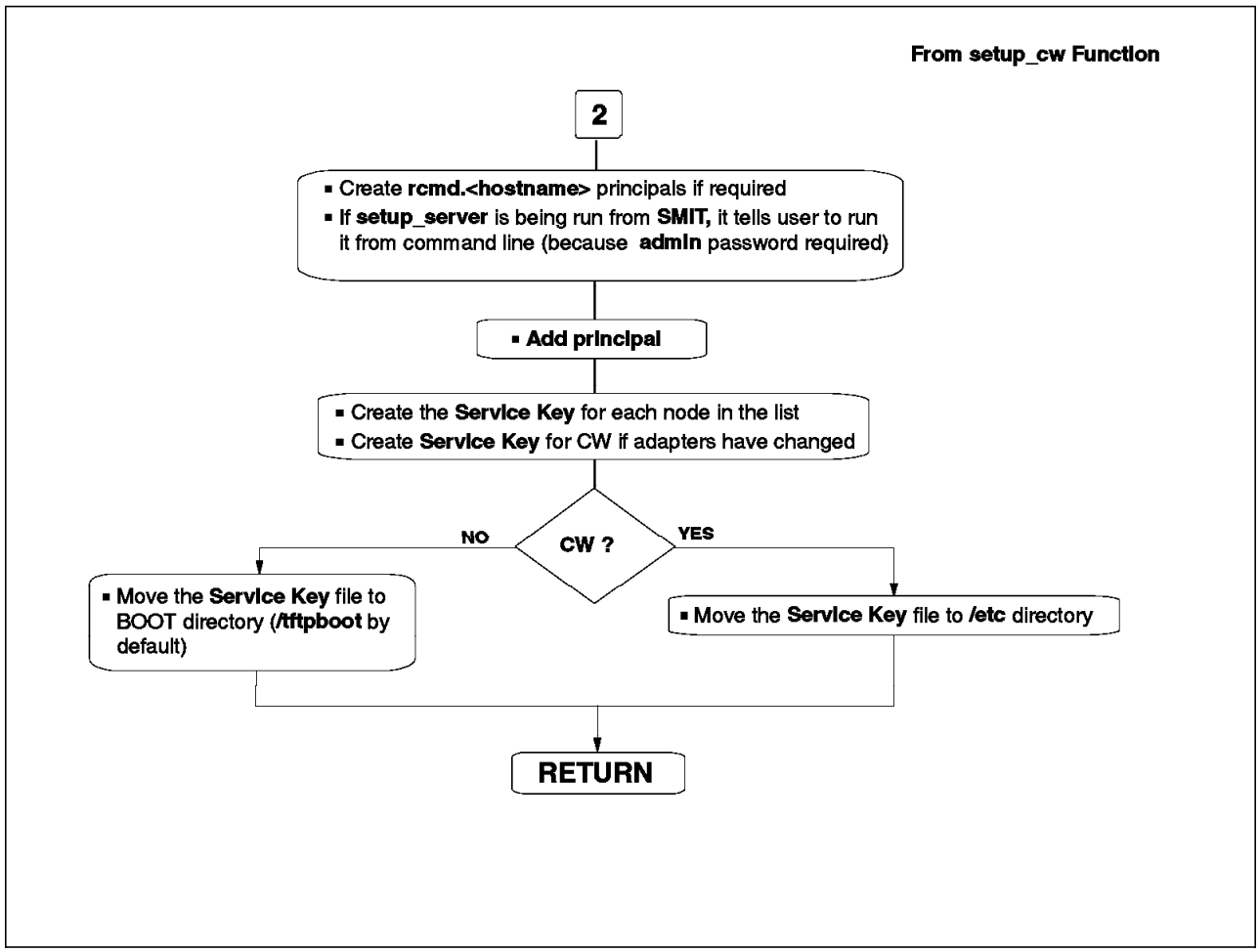


Figure 119. setup_server Script Flow Chart (9/23)

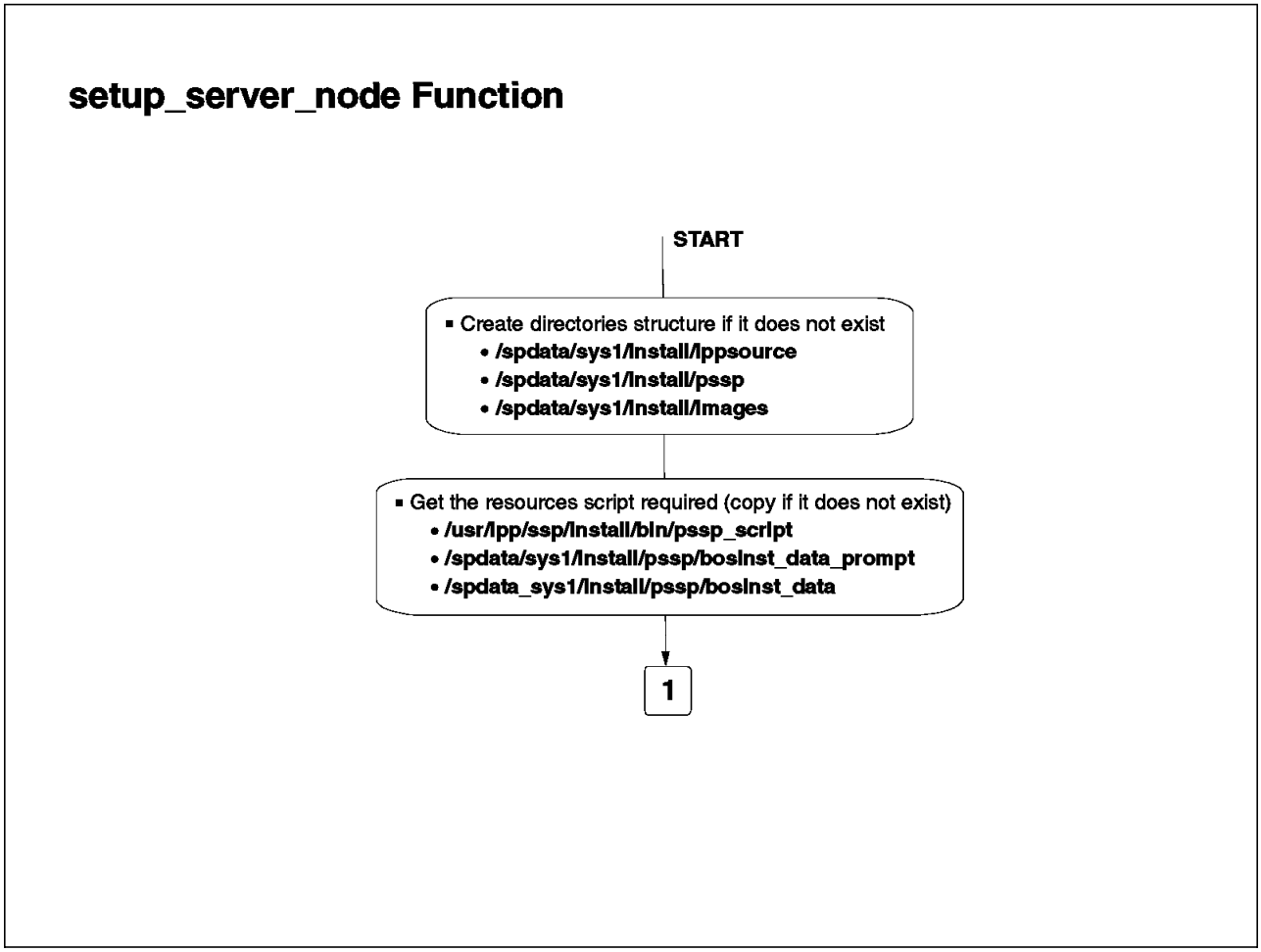


Figure 120. setup_server Script Flow Chart (10/23)

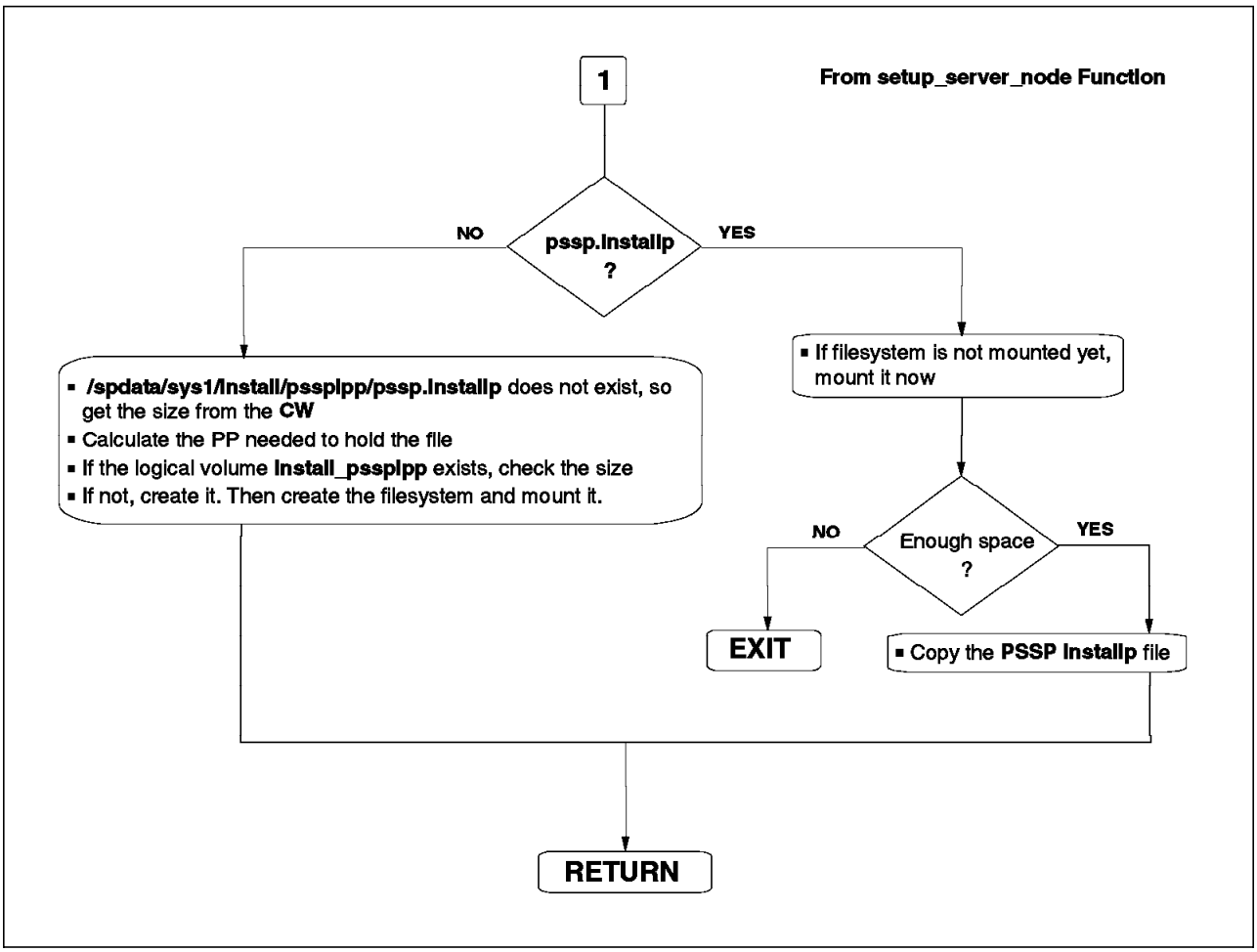


Figure 121. setup_server Script Flow Chart (11/23)

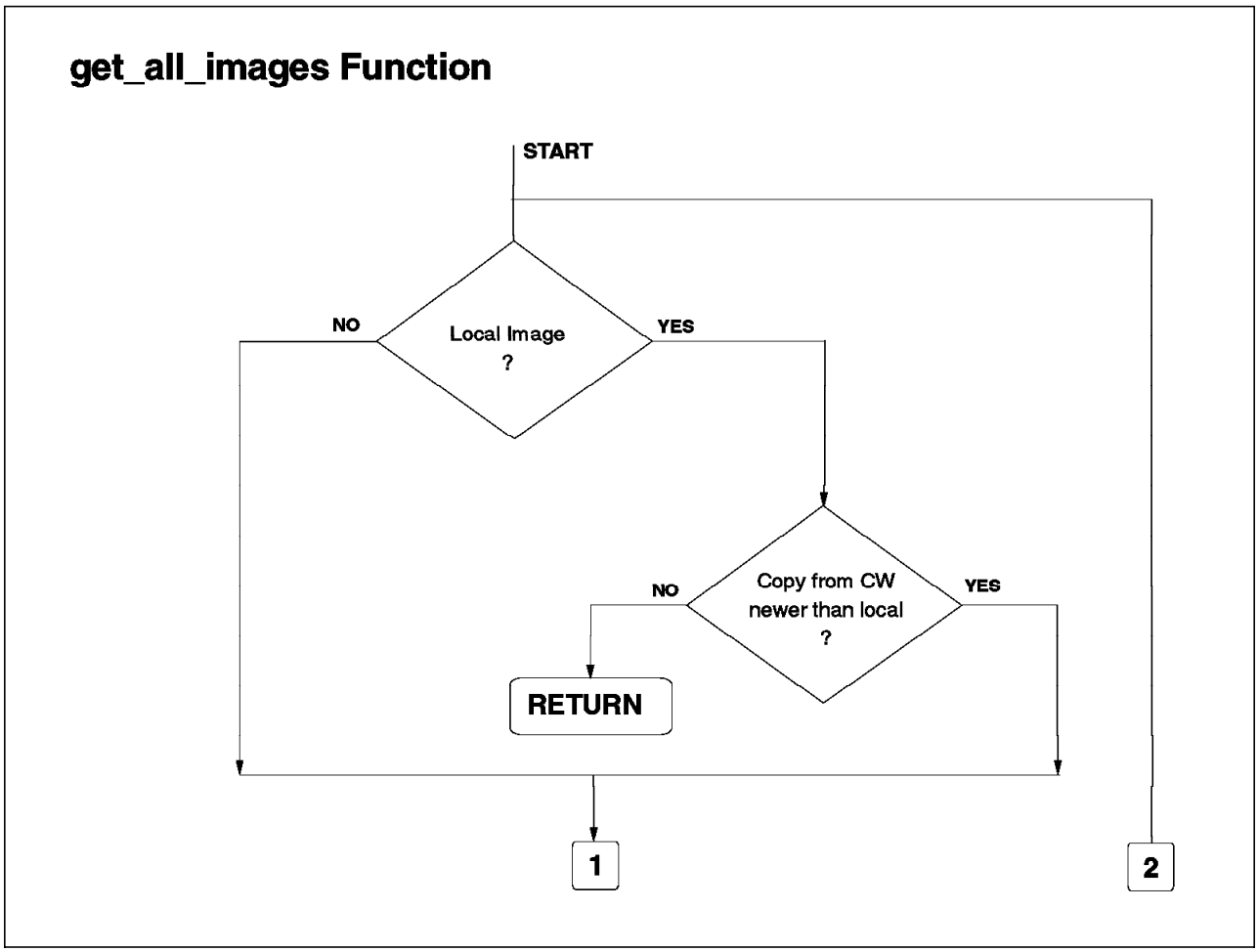


Figure 122. setup_server Script Flow Chart (12/23)

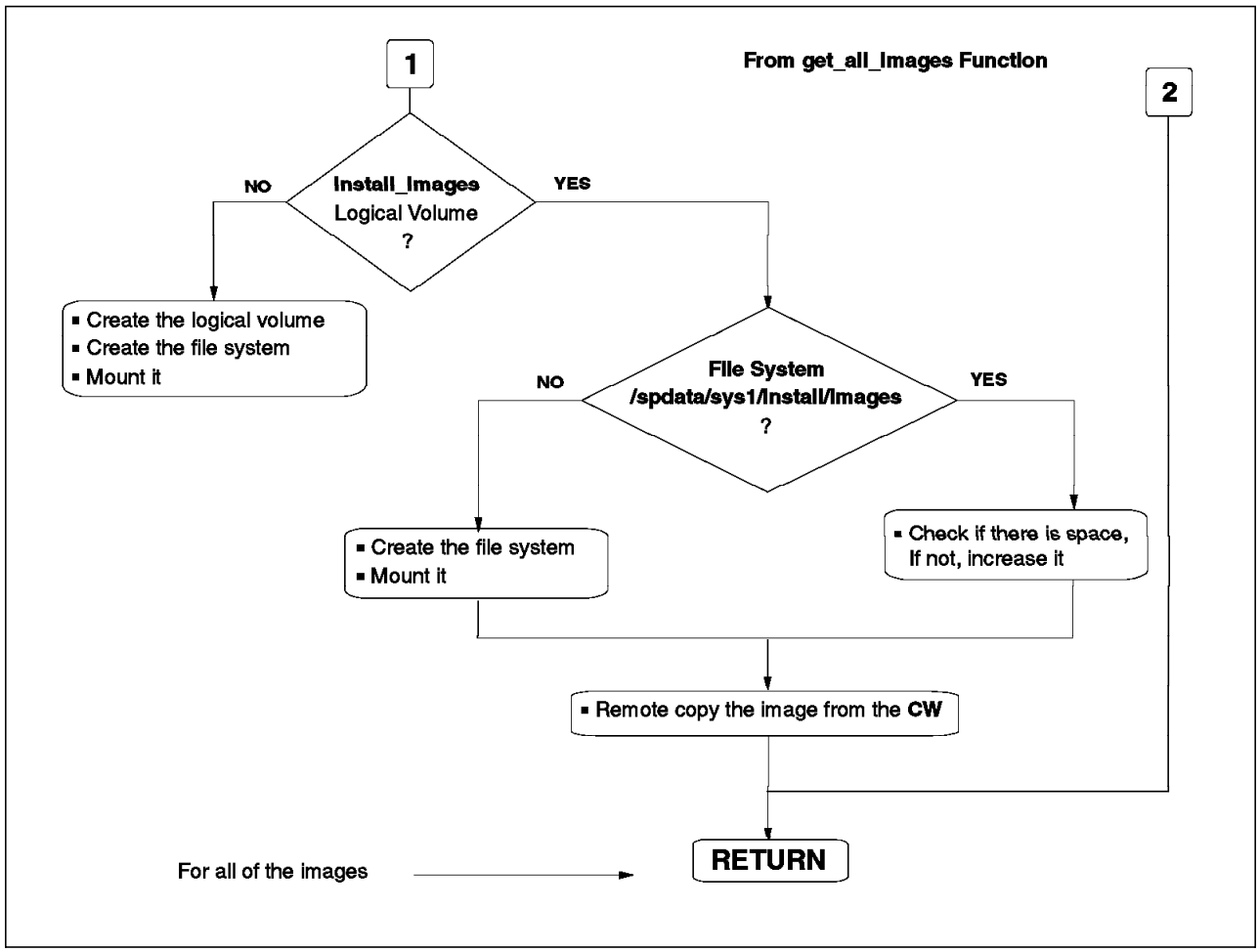


Figure 123. setup_server Script Flow Chart (13/23)

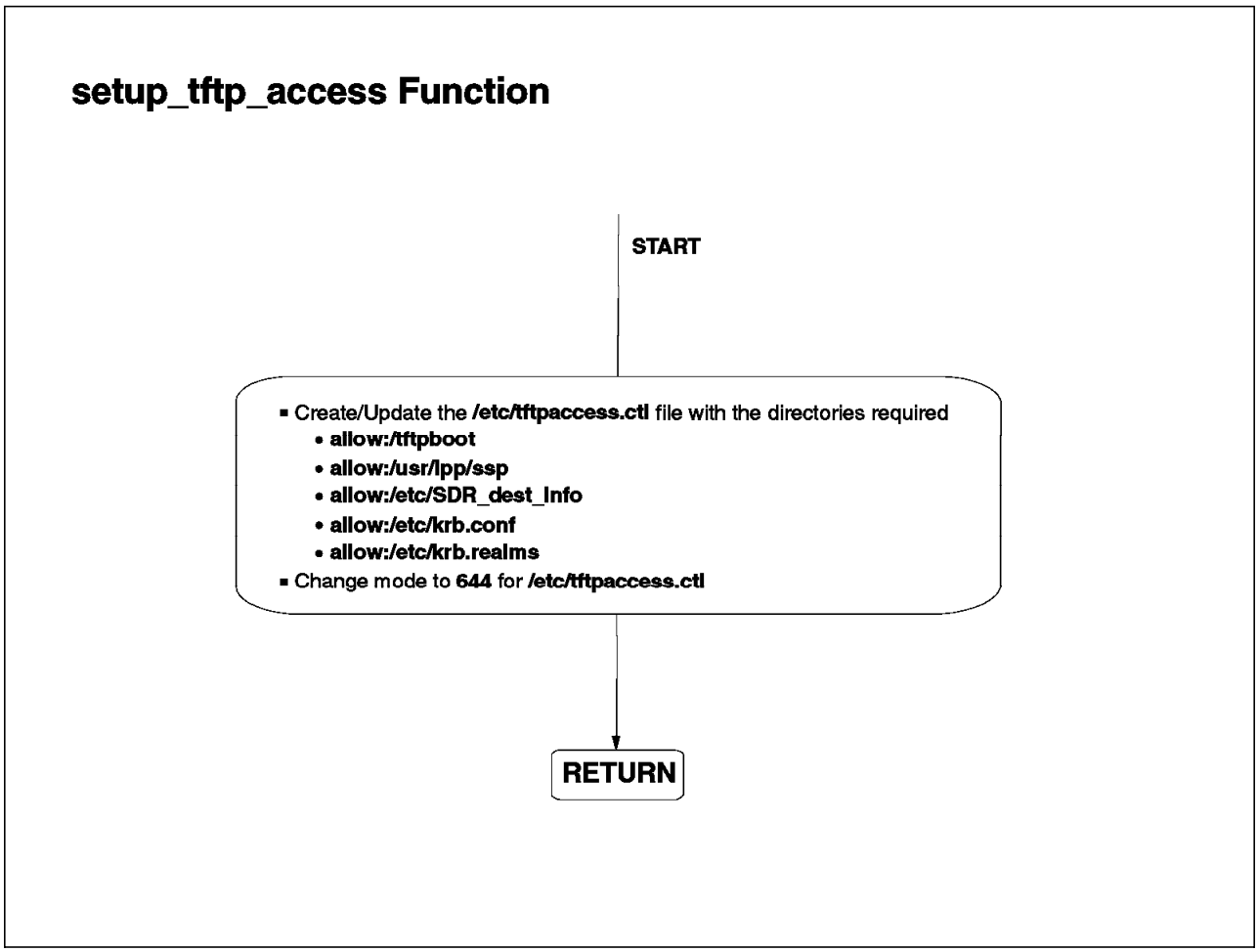


Figure 124. setup_server Script Flow Chart (14/23)

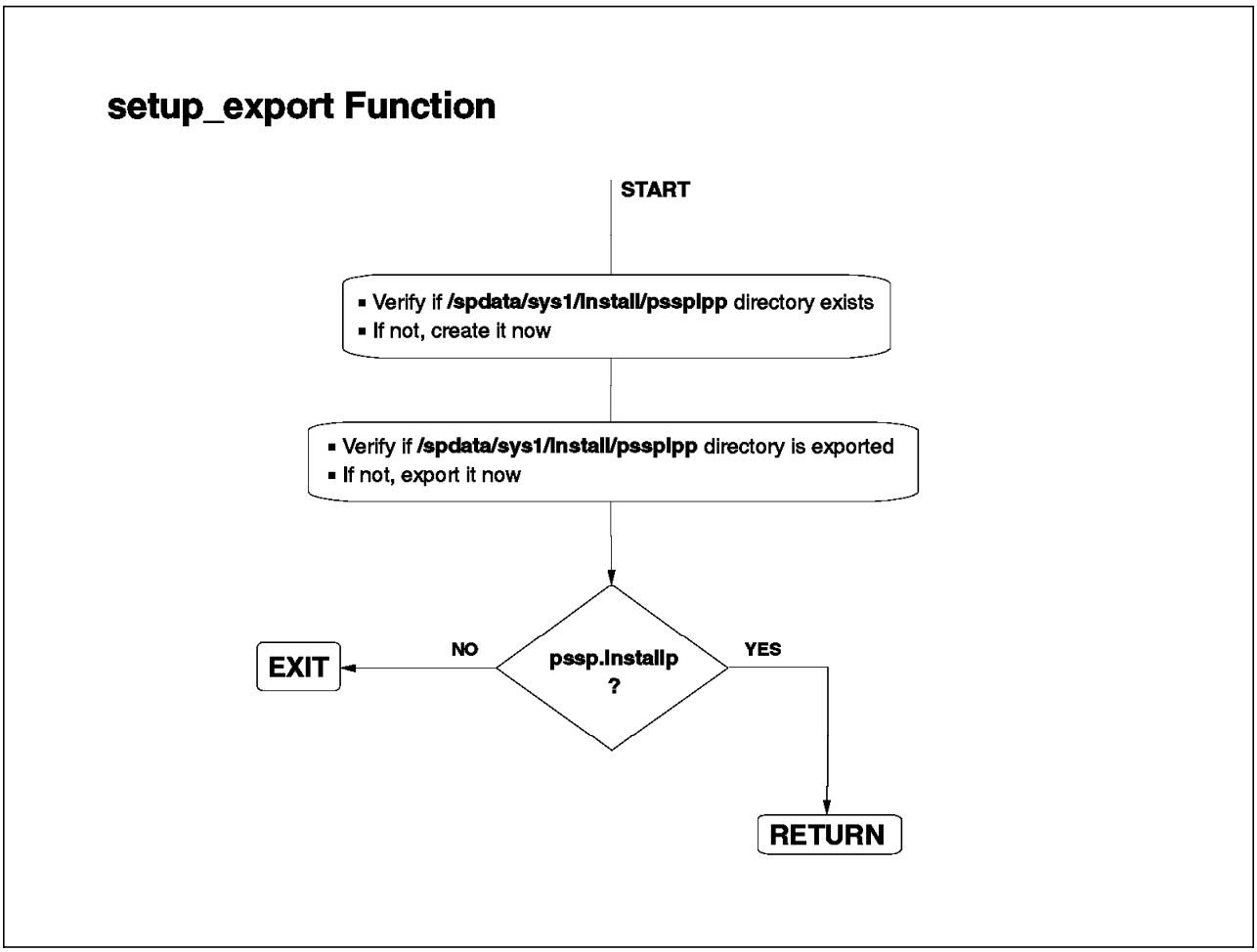


Figure 125. setup_server Script Flow Chart (15/23)

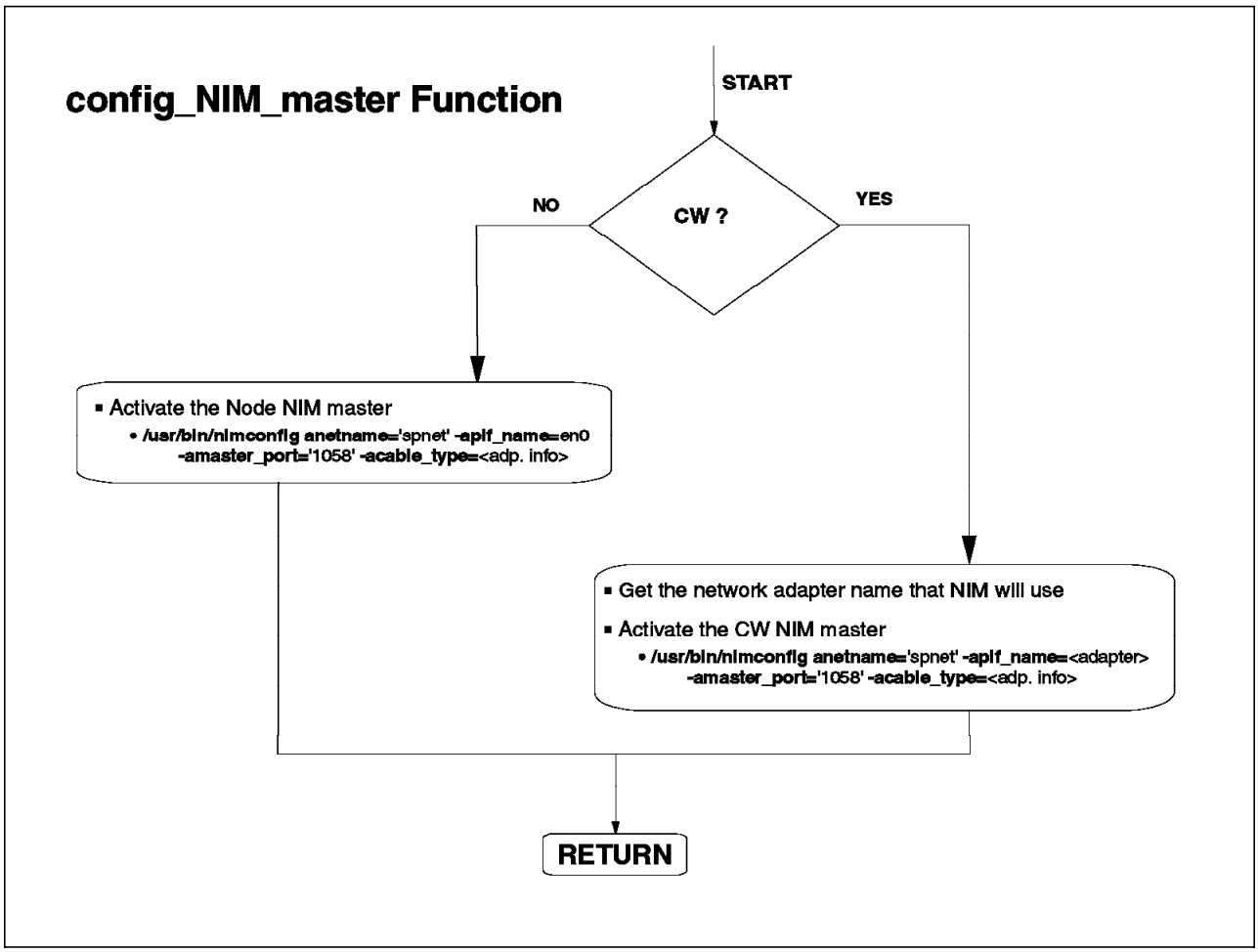


Figure 126. setup_server Script Flow Chart (16/23)

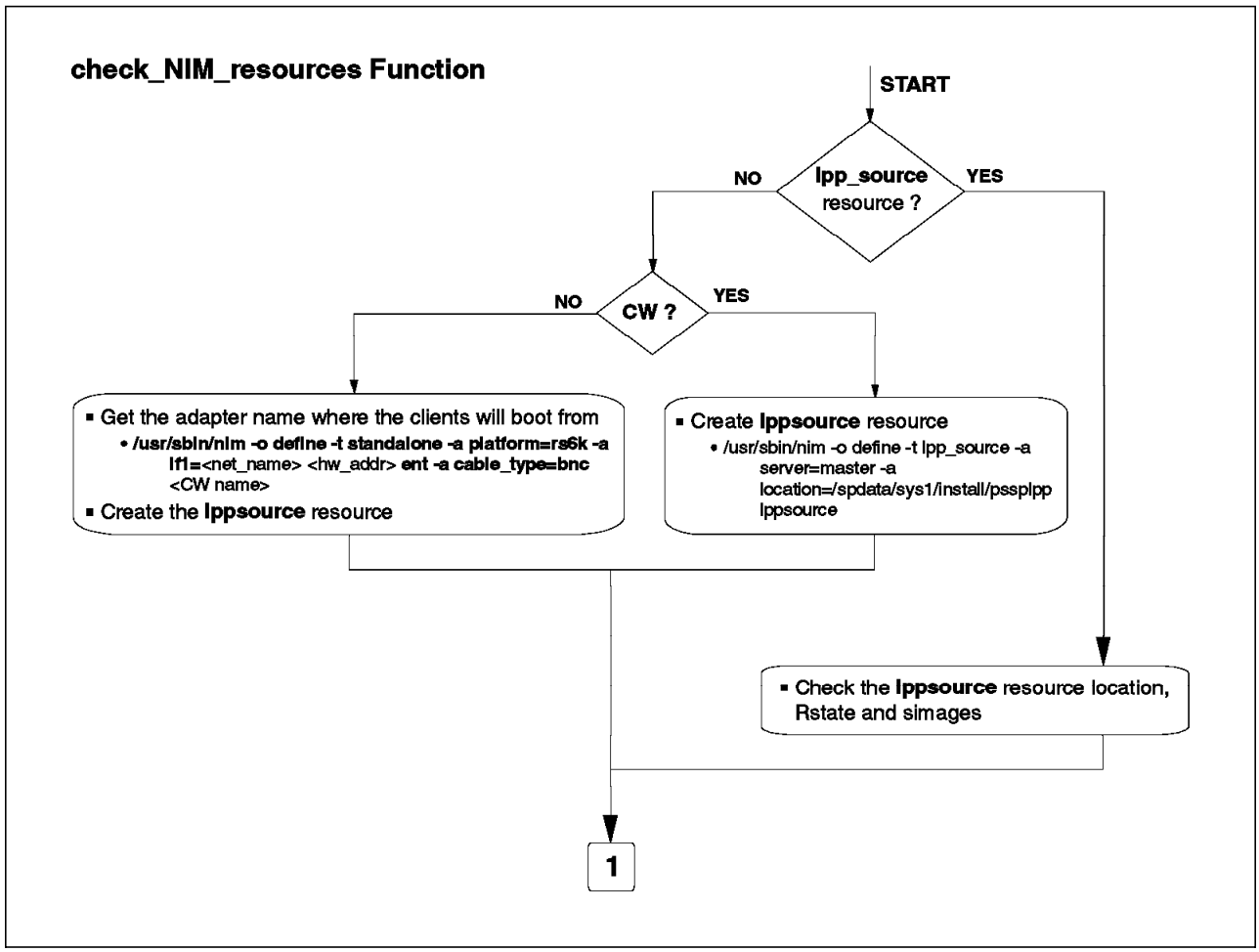


Figure 127. setup_server Script Flow Chart (17/23)

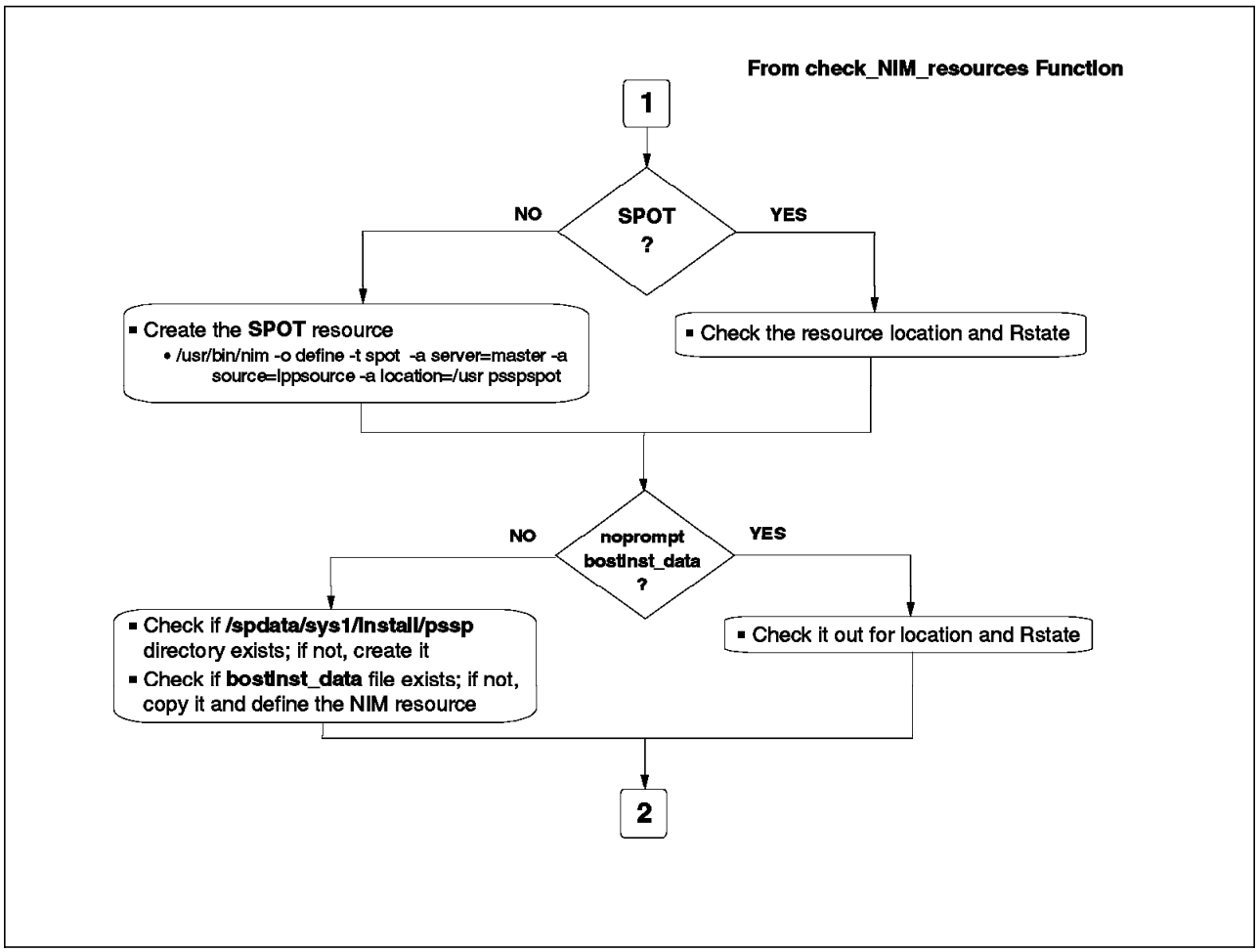


Figure 128. setup_server Script Flow Chart (18/23)

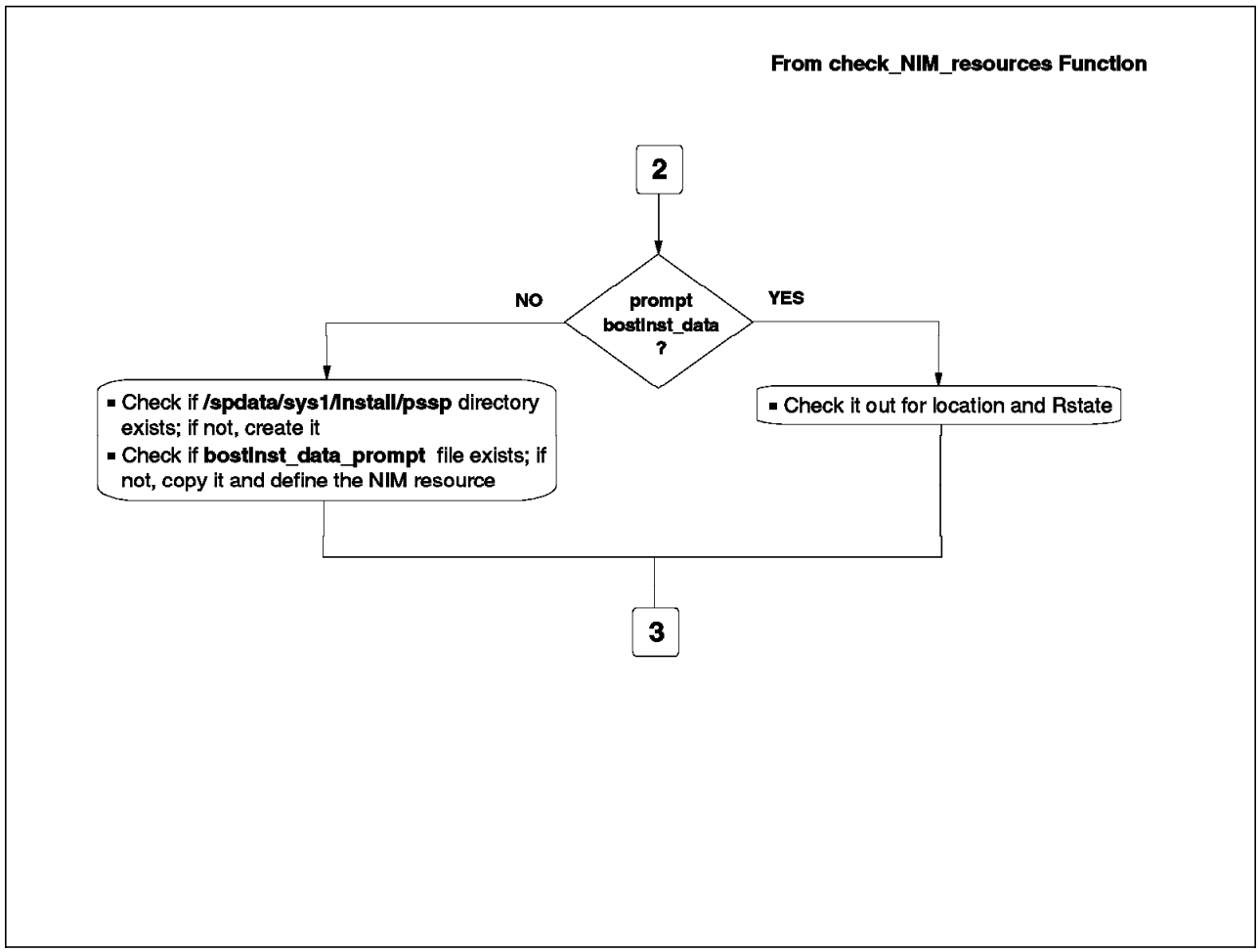


Figure 129. setup_server Script Flow Chart (19/23)

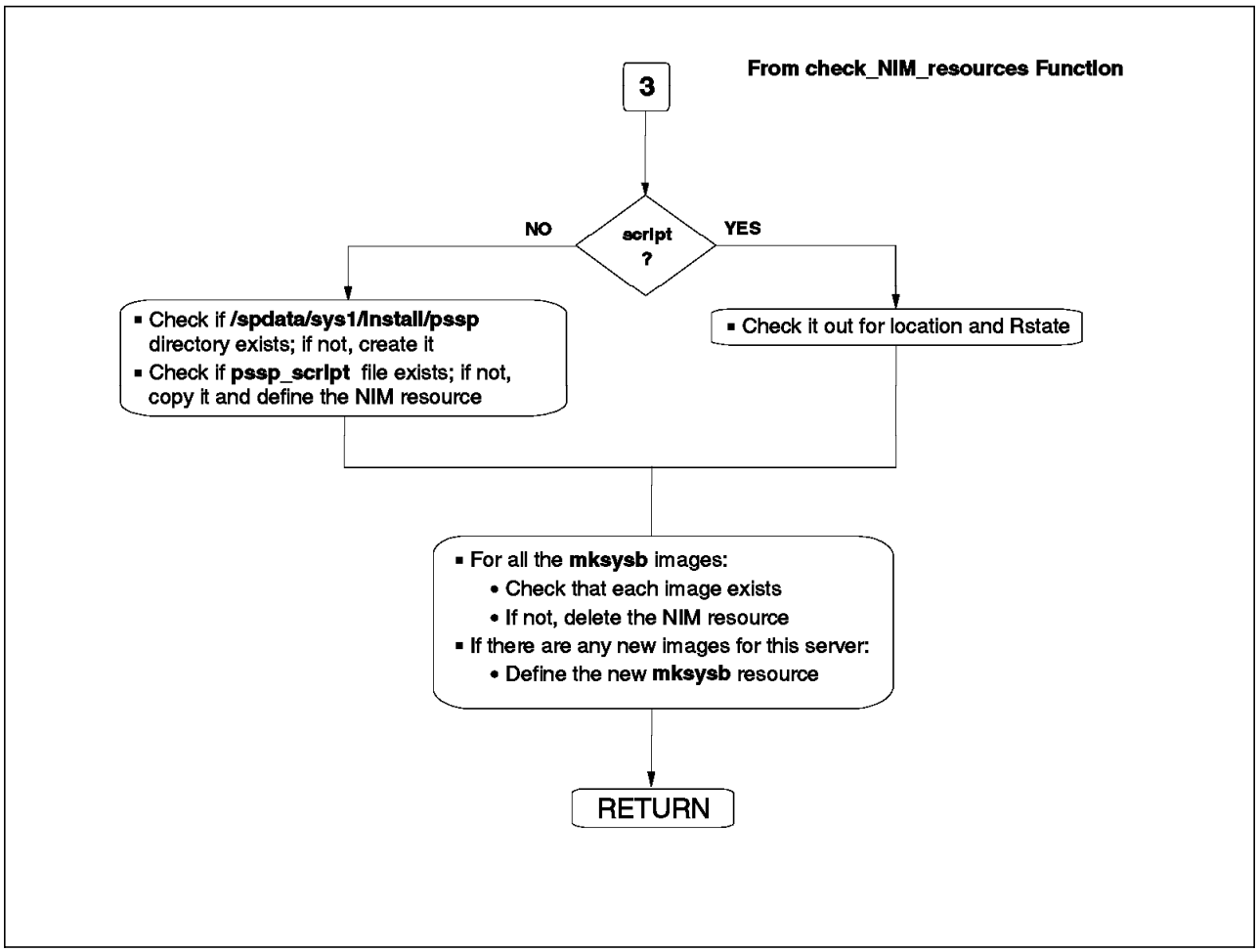


Figure 130. setup_server Script Flow Chart (20/23)

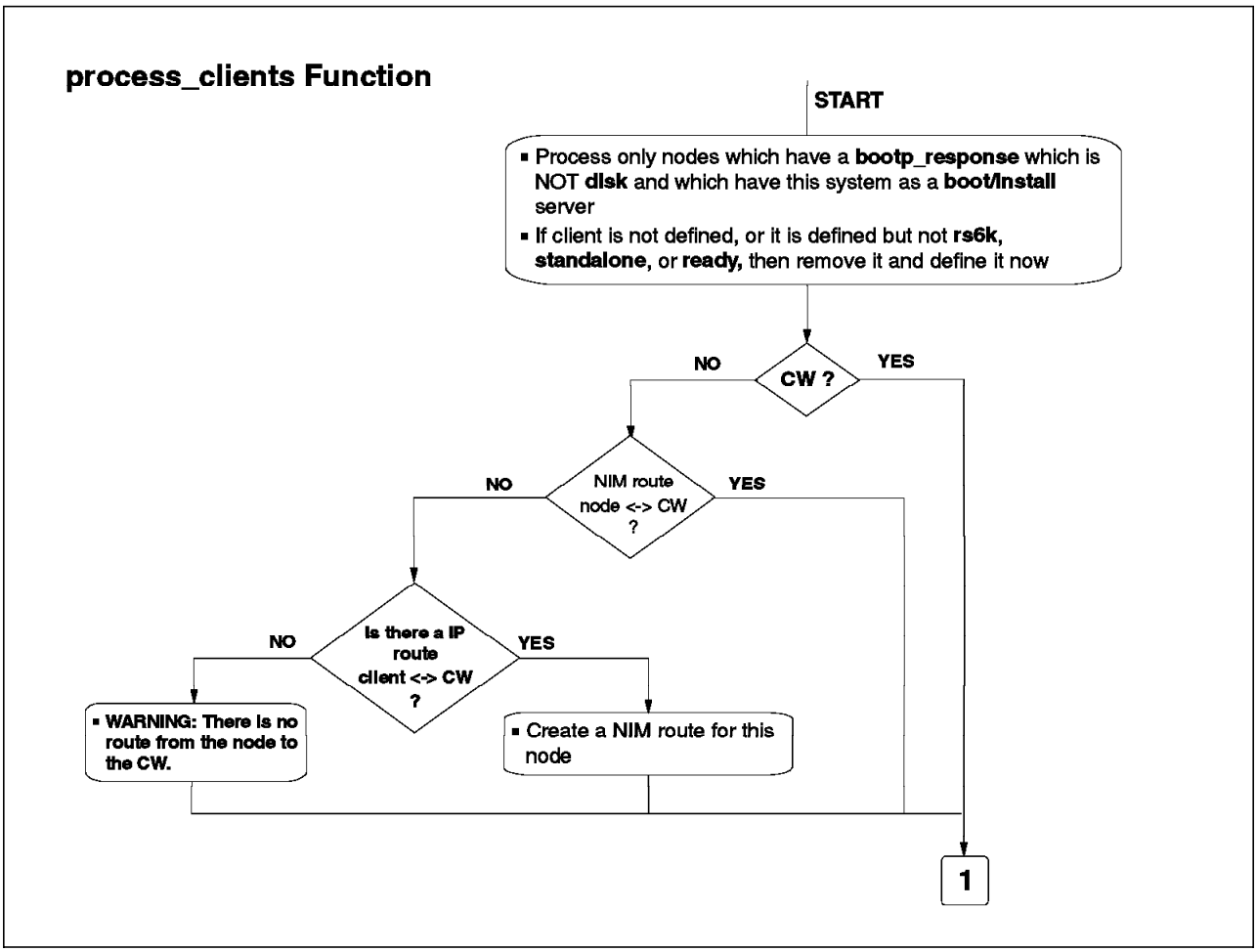


Figure 131. setup_server Script Flow Chart (21/23)

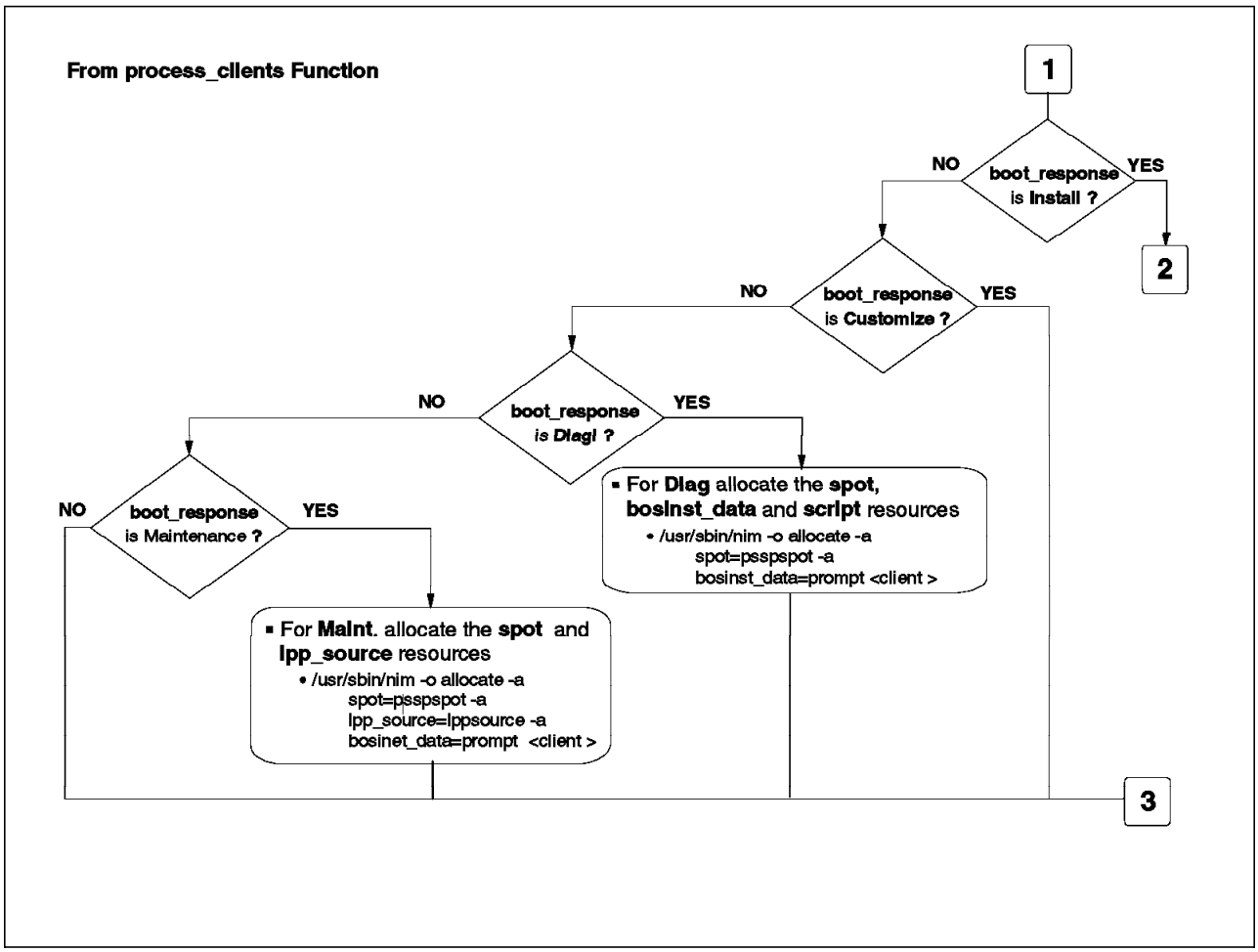


Figure 132. setup_server Script Flow Chart (22/23)

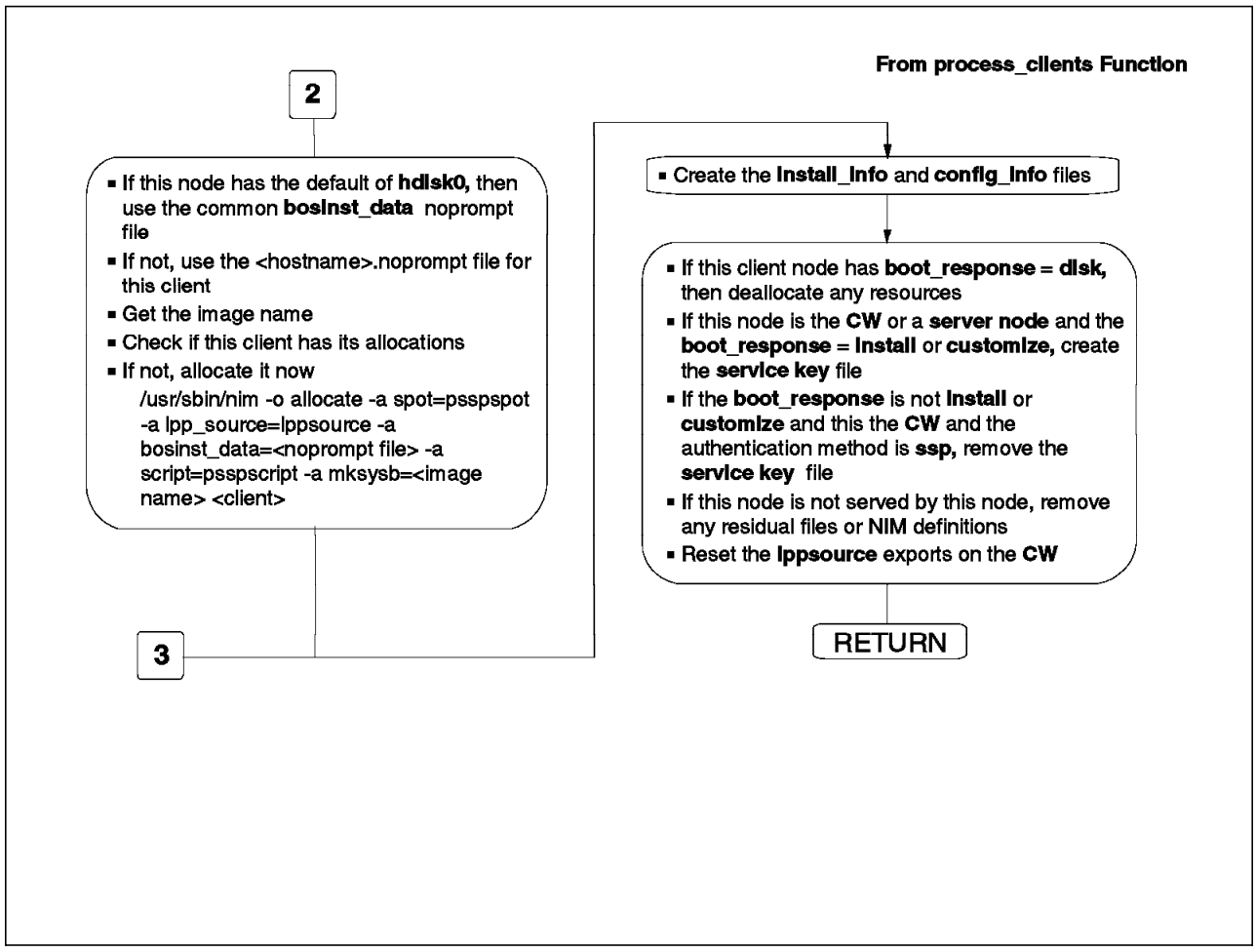


Figure 133. setup_server Script Flow Chart (23/23)

A.4 The rc.switch Script

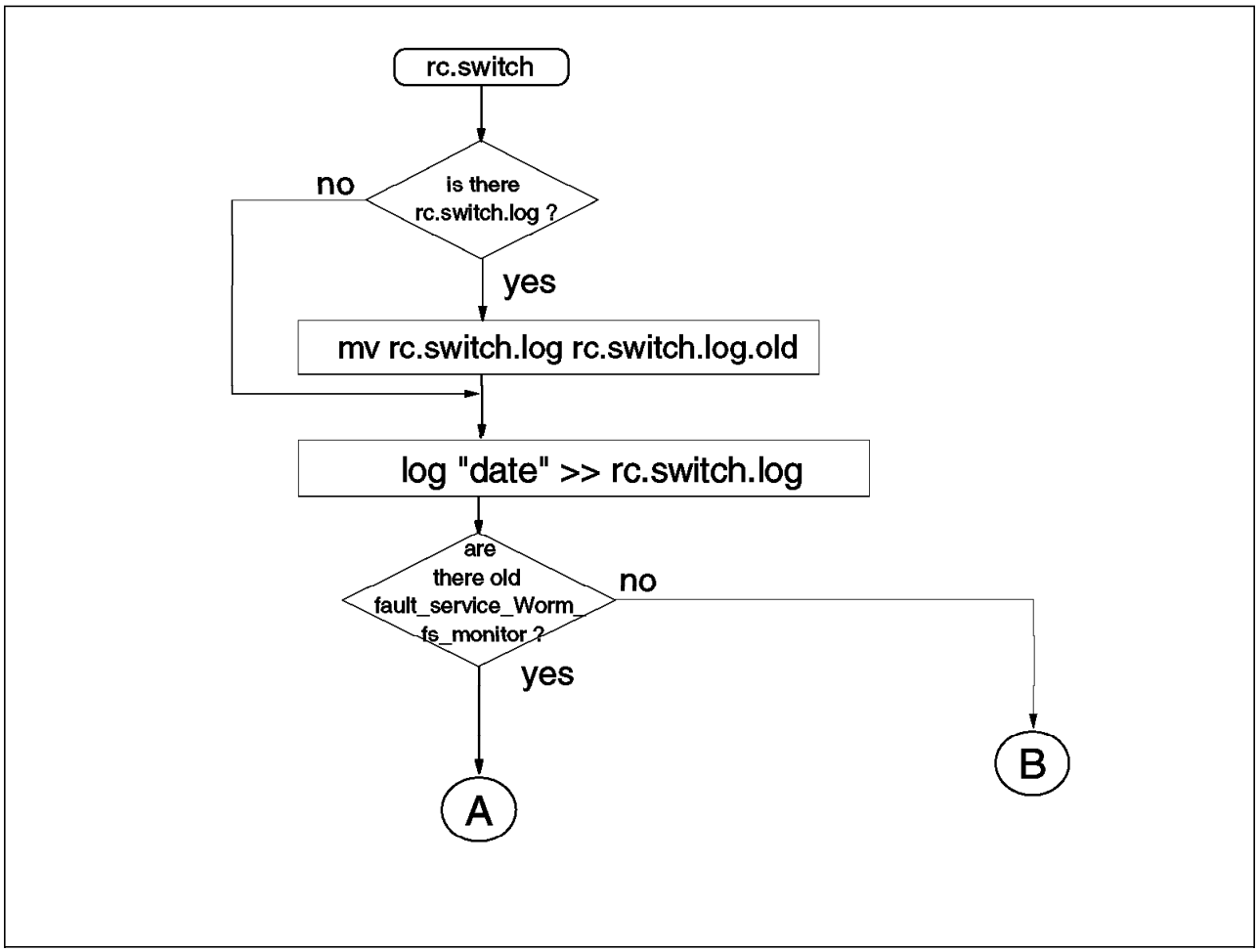


Figure 134. rc.switch Script Flow Chart (1/8)

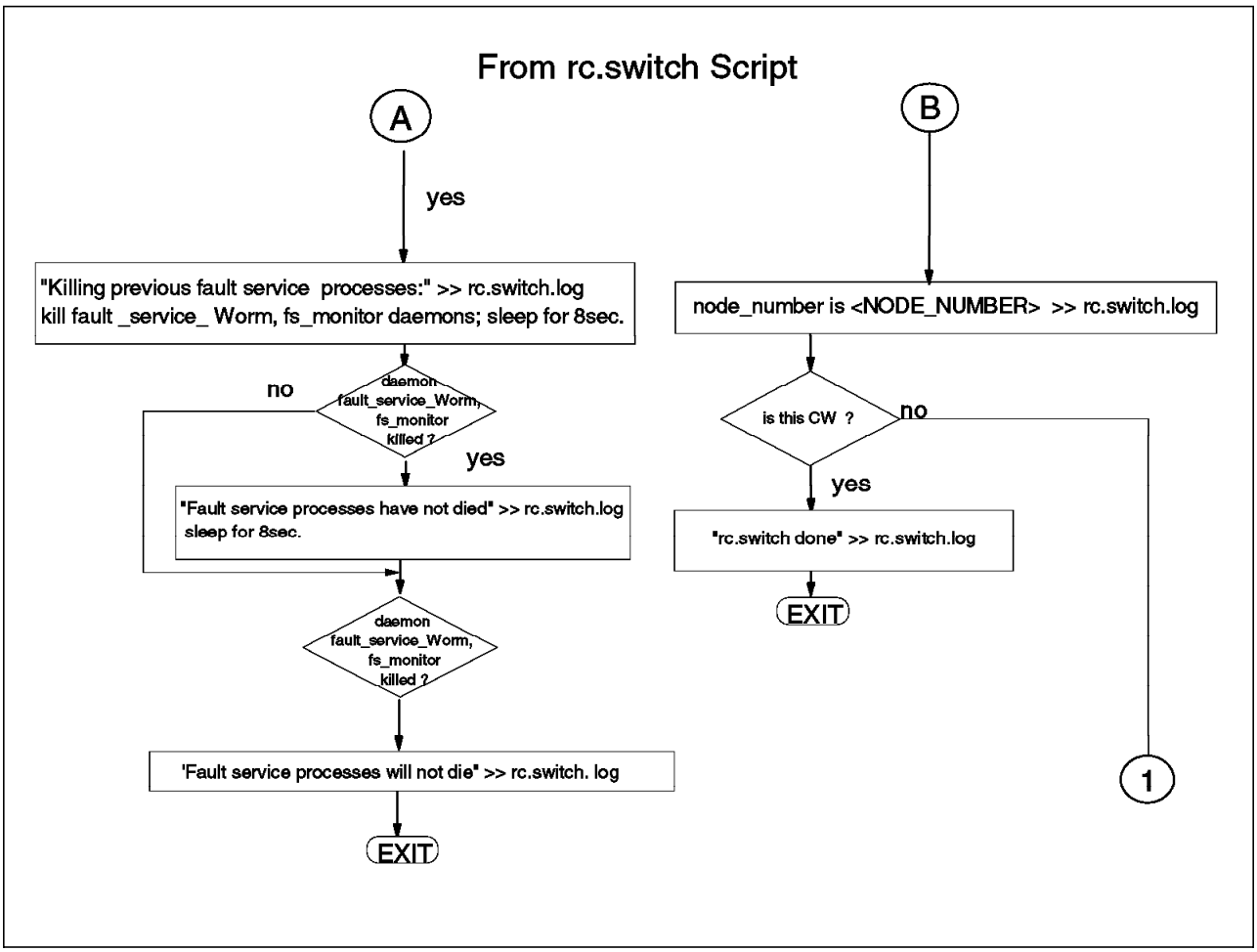


Figure 135. rc.switch Script Flow Chart (2/8)

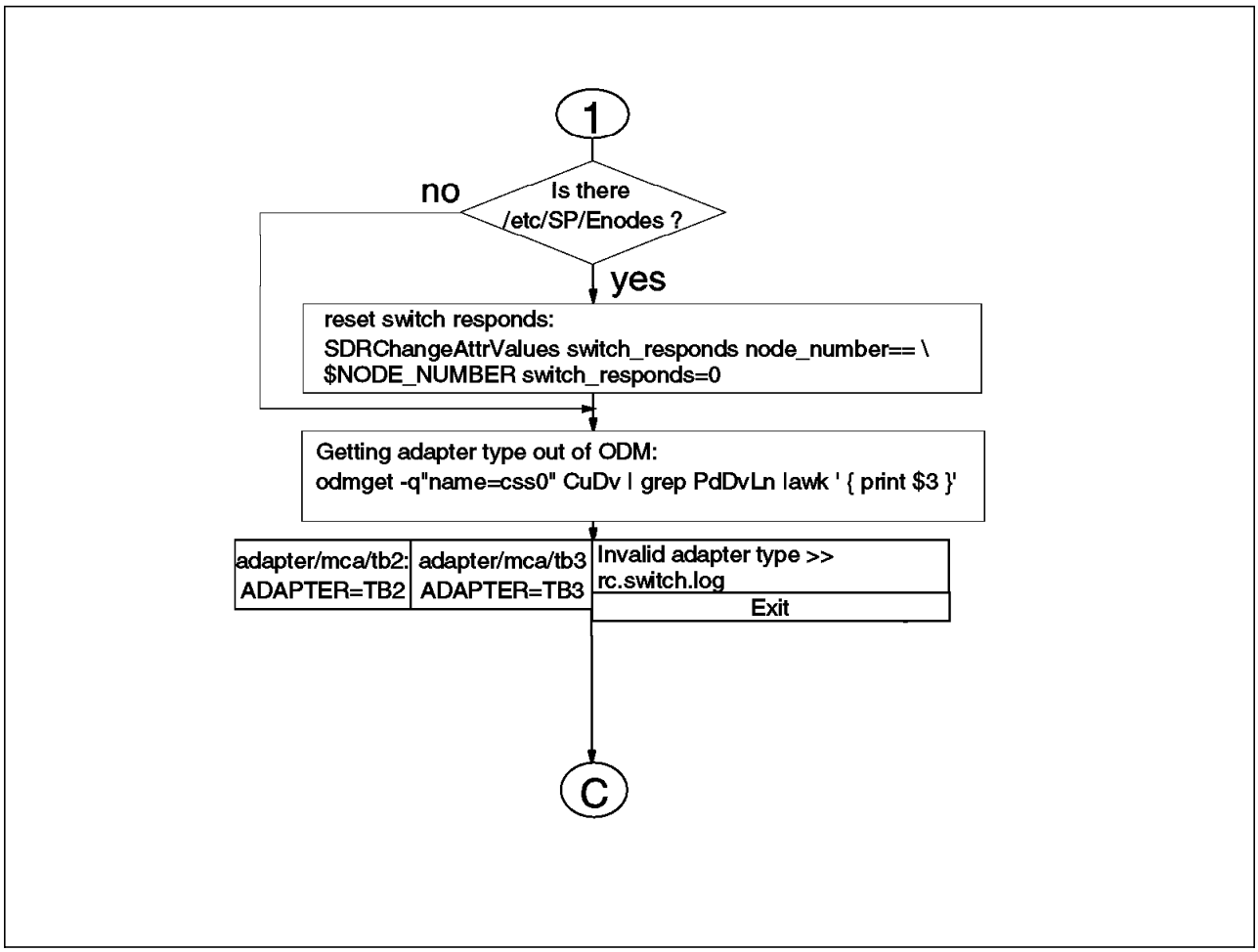


Figure 136. rc.switch Script Flow Chart (3/8)

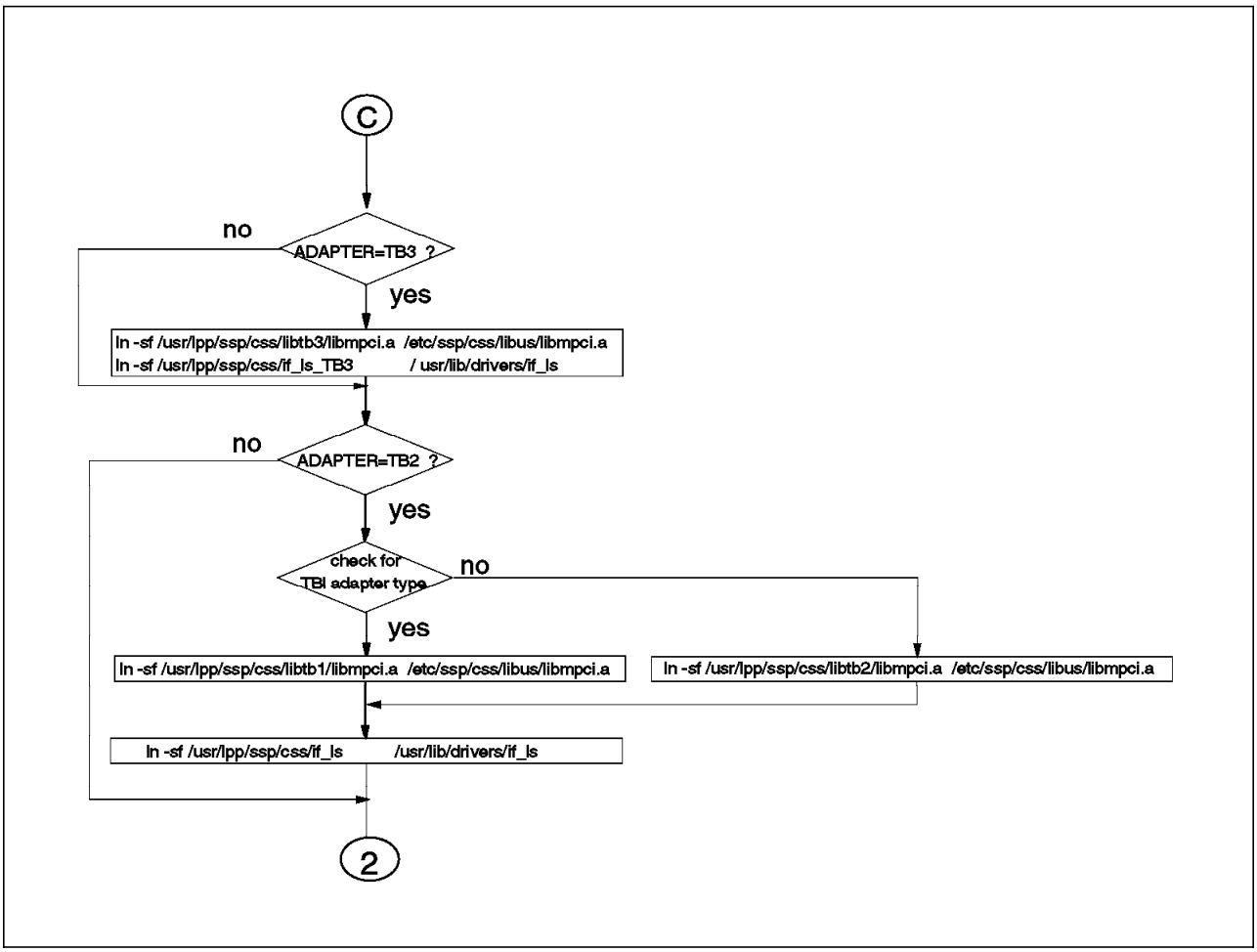


Figure 137. rc.switch Script Flow Chart (4/8)

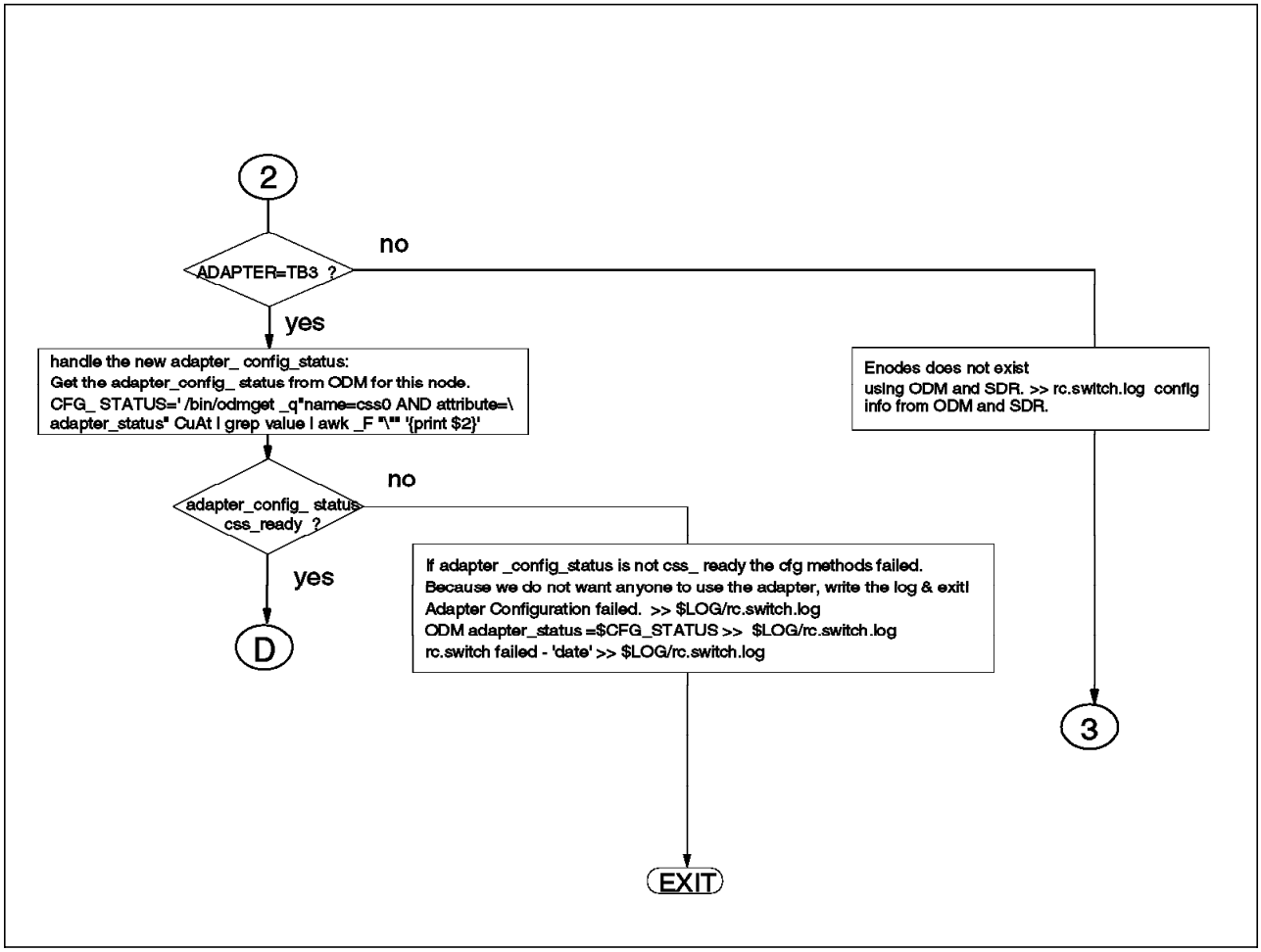


Figure 138. rc.switch Script Flow Chart (5/8)

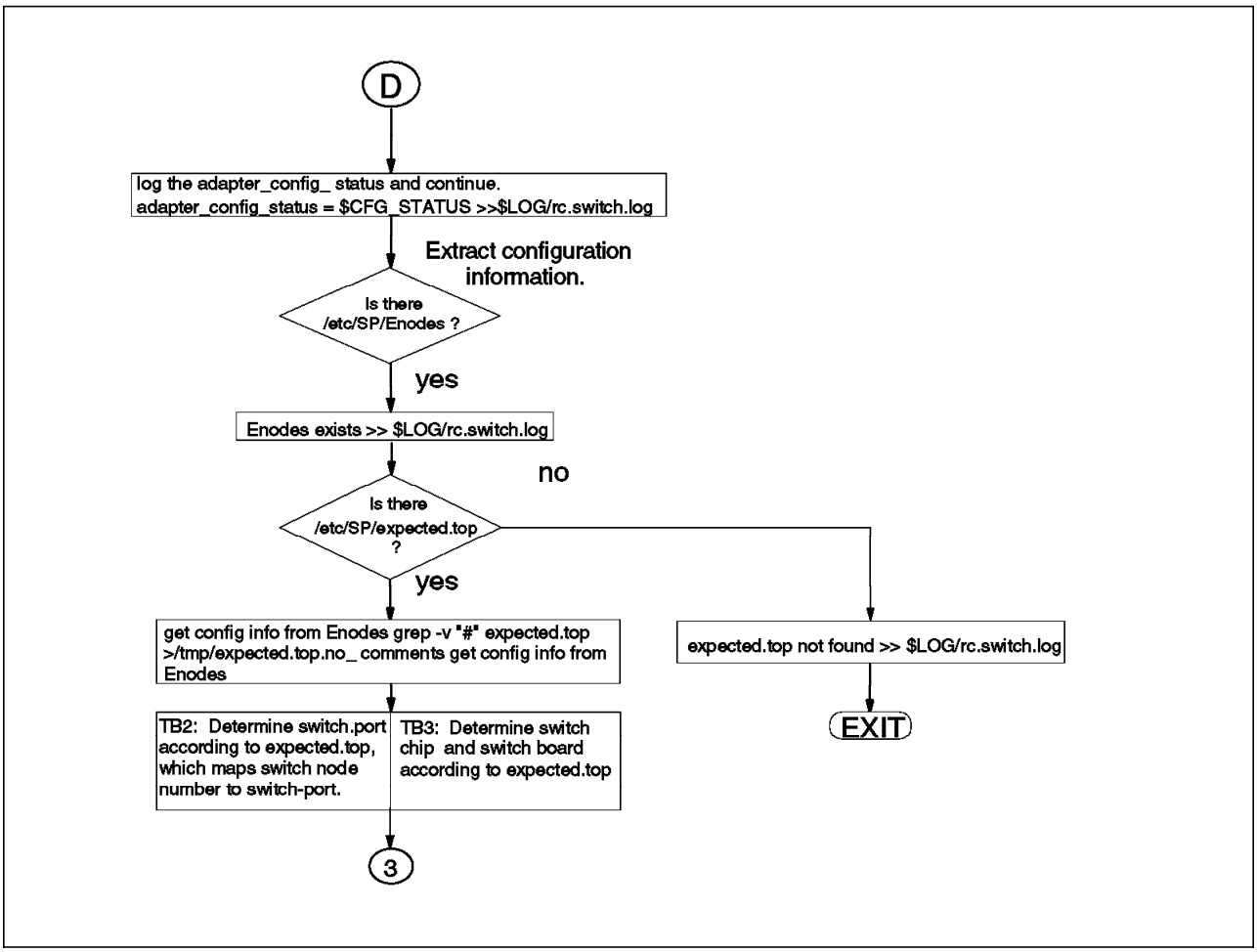


Figure 139. rc.switch Script Flow Chart (6/8)

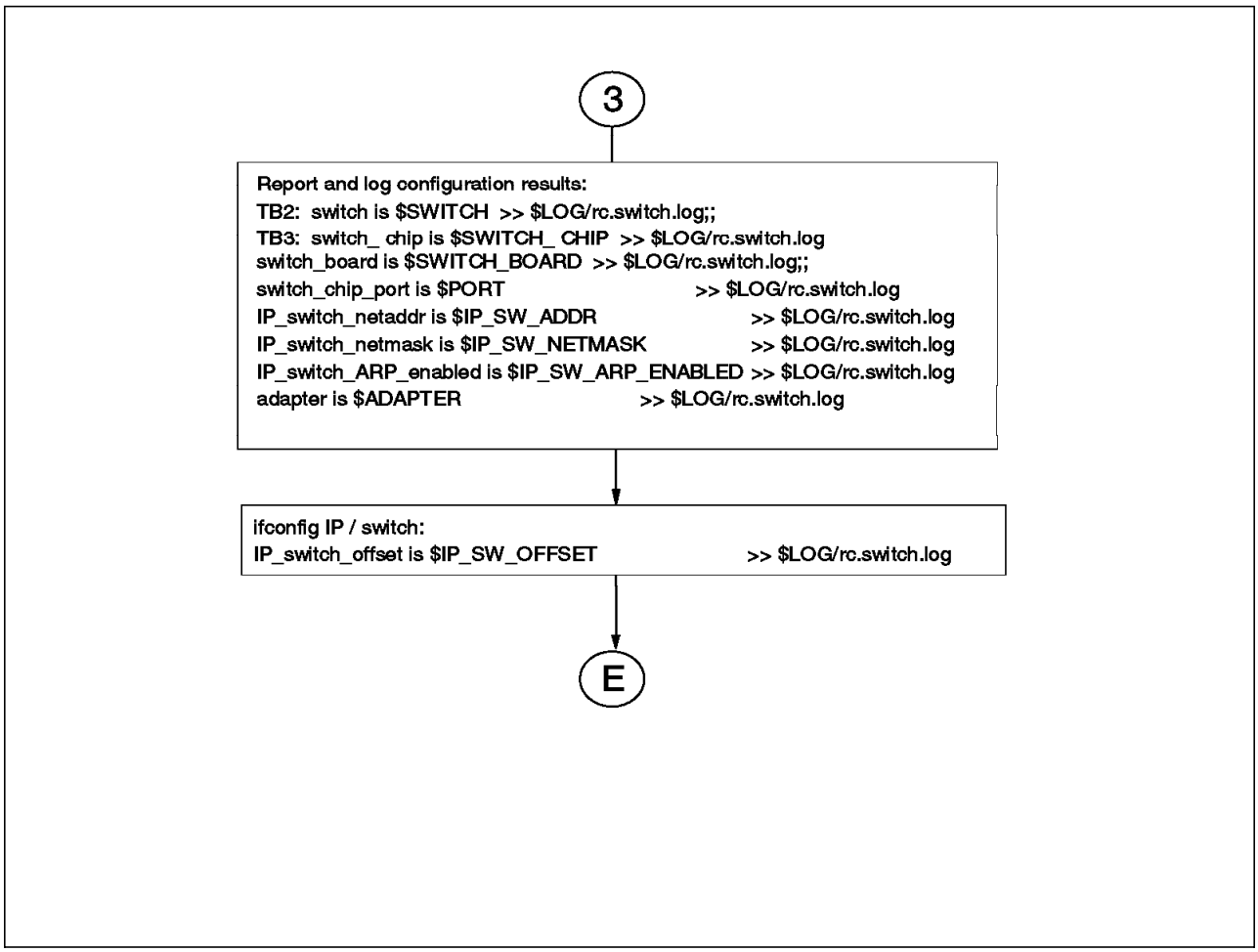


Figure 140. rc.switch Script Flow Chart (7/8)

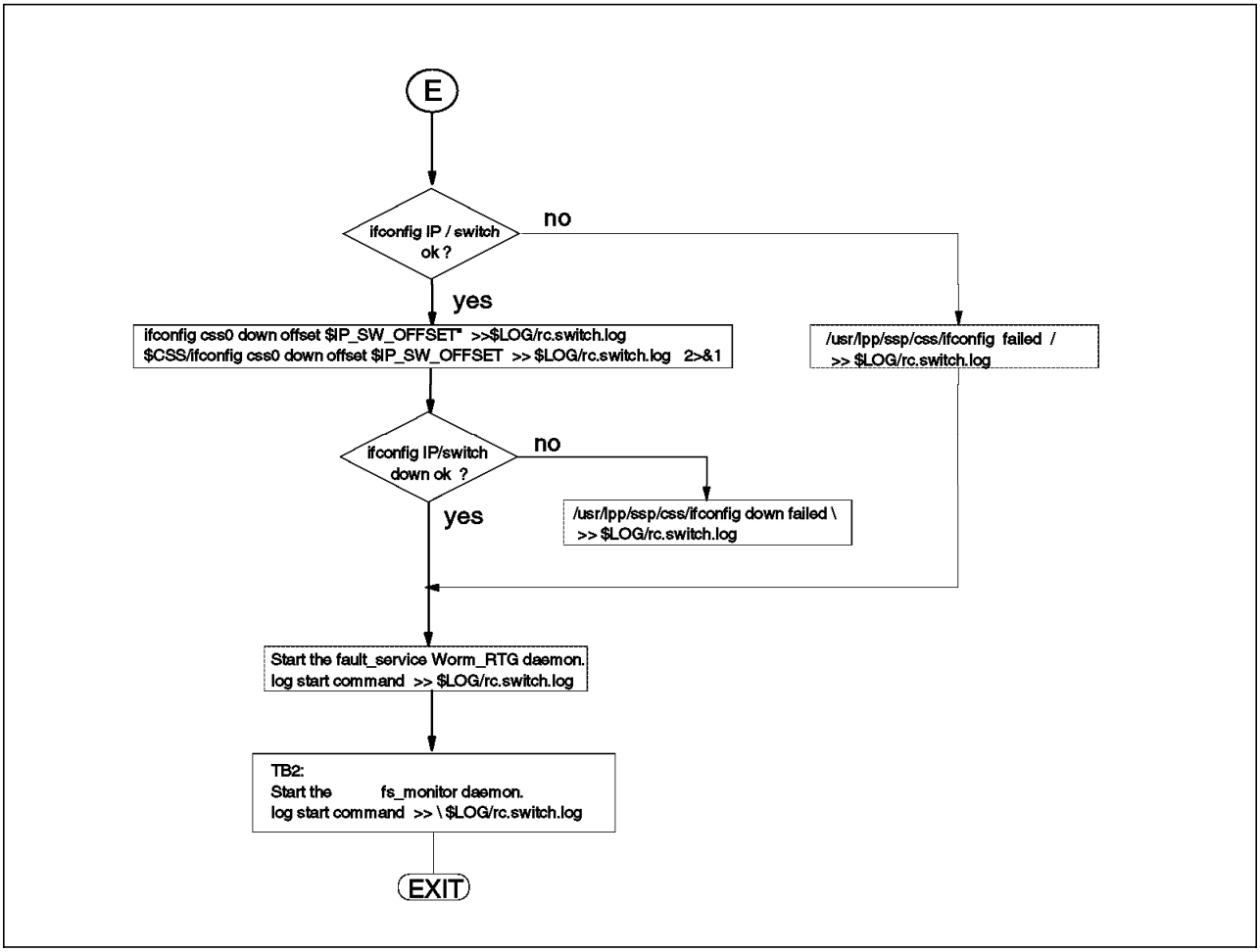


Figure 141. rc.switch Script Flow Chart (8/8)

Appendix B. The SDR Structure

The SDR is the central data repository for the RS/6000 SP. The communication with the SDR is through a client/server relationship. The server is represented by a process (daemon) running on the Control Workstation, the client portion is represented by different SP commands, and it is also represented by SP subsystems that query the SDR to retrieve information about the SP configuration or values for state variables.

The organization of the SDR database is based on object classes. Each object class contains different attributes, and there can be many instances of a particular object class.

When the SP is partitioned, some of the SDR classes are also partitioned. Each partition will have a different SDR process (daemon) who will manage the SDR database portion that has been partitioned.

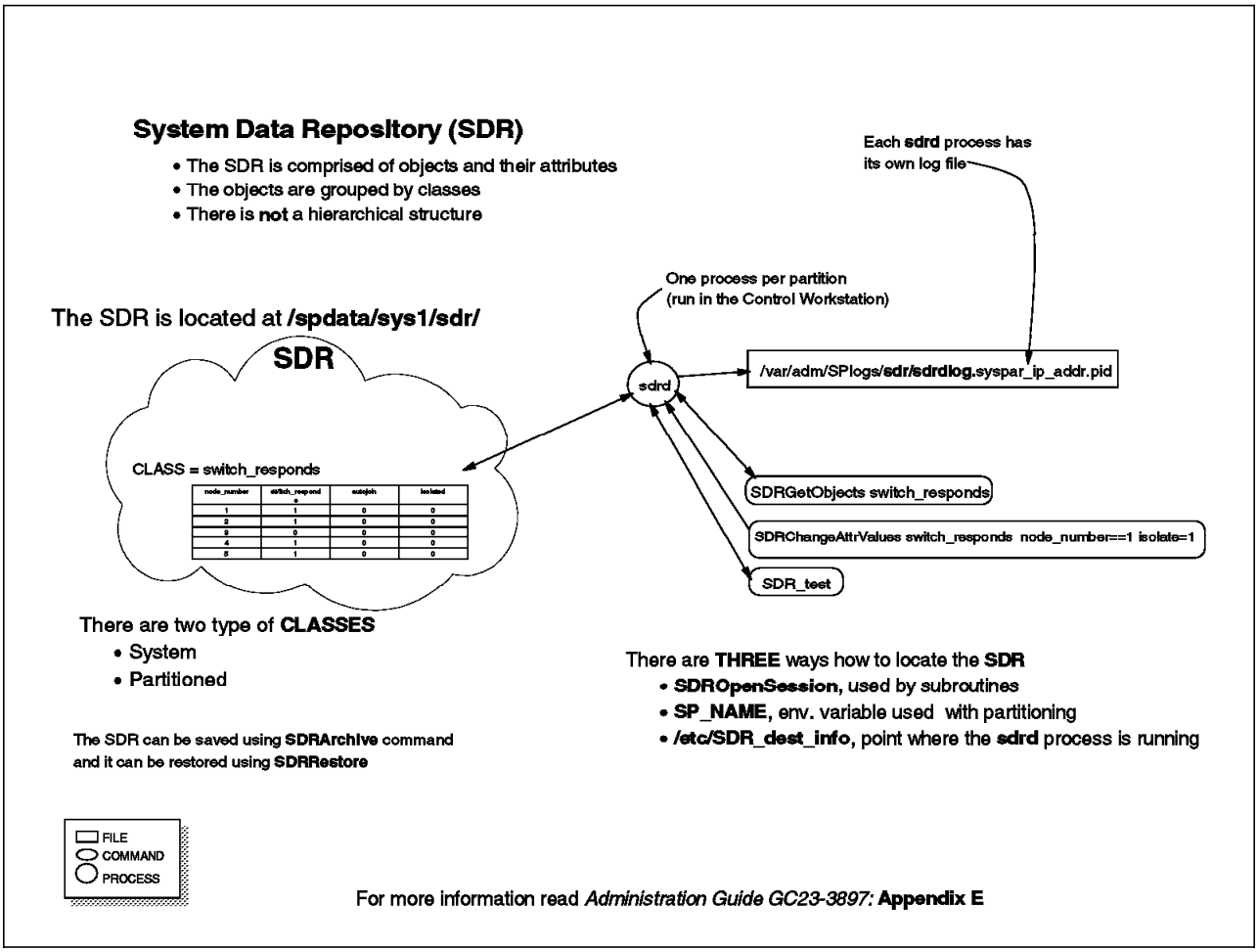


Figure 142. The SDR Structure

Appendix C. IP Address and Hostname Changes for the SP

This information describes the activities that the admin user needs to execute when changing the IP address or hostname for the RS/6000 SP nodes and Control Workstation (CW). It is highly recommended to avoid making any IP address or hostname changes if possible. The tasks are strenuous and in some cases require re-execution of the installation steps.

The changing of the IP addresses/hostnames of RS/6000 SP nodes affect the whole RS/6000 SP system. IP addresses/hostnames are located in the System Data Repository (SDR) working with objects and attributes.

IP addresses/hostnames are also kept in various files that are located on the RS/6000 SP nodes and the CW. These changes will also cause problems with Network Installation Manager (NIM) and System Partition objects and configuration files on your RS/6000 SP system.

C.1 SDR Objects

The SDR objects that reference the hostname and IP address changes are :

Adapters	Deals with ent, css0, tr0, fi0 IP addresses and adapters.
Frame	Deals with the MACN CWS attribute and works with hostnames.
Node	Works with initial/reliable hostnames and uses the IP address for boot servers. Node objects are organized in system partitions.
SP	Deals mostly with CW attributes, but uses hostname when working with ntp, printing, user management, and accounting.
SP_ports	Deals with hostname used with hardmon and CW.
Switch_partition	Works with hostname of primary node used with switch.
JM_domain,Pool	Works with Resource Manager and references hostnames.
Syspar	Deals with IP address and SP_NAME used with system partitions.

C.2 RS/6000 SP System Files

The following are files that contain IP addresses, or hostnames that exist on RS/6000 SP nodes and the CW. It is best to look through these files when completing the procedures for changing hostnames and IP addresses for your RS/6000 SP system.

/.rhosts	Contains hostnames used exclusively with rcmd services
/.klogin	Contains hostnames used with authentication rcmd services

/etc/hosts	Contains IP addresses and hostnames used with RS/6000 SP system
/etc/resolv.conf	Contains IP address for domain name service (Optional)
/etc/rc.net	Contains alias IP addresses used with system partitions
/var/yp/<nis>	Network Information Service references IP address/hostname
/etc/niminfo	Works with the NIM configuration for NIM master information
/etc/krb.conf	Works with hostname for authentication server
/etc/krb.realms	Works with hostname of RS/6000 SP nodes and authentication server
/etc/krb-srvtab	Provides authentication service key using hostname
/etc/SDR_dest_info	Specifies hostname and IP address of the CW and SDR
/etc/ssp/cw_name	Specifies the IP address of the CW hostname (RS/6000 SP nodes)
/etc/ssp/server_name	Specifies IP address and hostname of RS/6000 SP server of the boot/install adapter servers (RS/6000 SP nodes)
/etc/ssp/server_hostname	Specifies IP address and hostname of RS/6000 SP server
/etc/ssp/reliable_hostname	Specifies IP address and hostname of RS/6000 SP node
/etc/ntp.conf	Works with IP address of the ntp server (Optional)
/etc/filesystems	May contain IP address or hostname of nfs systems mainly used on /usr client systems
/etc/jmd_config	Works with hostnames for Resource Management pools (CW)
/tftpboot/<host>.config_info	Contains IP address and hostname for each RS/6000 SP node, and is found on CW and boot servers
/tftpboot/<host>.intstall_info	Contains IP address and hostname for each RS/6000 SP node, and is found on CW and boot servers
/tftpboot/<host>-new-srvtab	Provides authentication service keys using hostname, and is found on CW and boot servers

C.3 Procedures Used When Changing IP Addresses/Hostnames

The changing of an IP address or hostname for a RS/6000 SP node and CW will follow the same procedures, except that the Control Workstation will include additional network and RS/6000 SP configuration steps.

You will need to update PSSP files and interfaces on the RS/6000 SP nodes. You will need to update the "en0" network interface on each RS/6000 SP node that has IP address, domain, or hostname changes. You will need to know the new selected IP addresses and hostnames being used by the RS/6000 SP nodes, Control Workstation, and gateway servers. This is required so that the RS/6000 SP node will properly communicate to the Control Workstation during the reboot activity. It will be necessary to do this activity manually for each RS/6000 SP node.

You will need to work on the Control Workstation and make the necessary changes to the System Data Repository (SDR). This includes updates to reset PSSP daemons, and system partitions (syspar) files. You need to update files on the Control Workstation to reflect the new hostnames and IP addresses. During this activity, the RS/6000 SP system should not be available to users. The RS/6000 SP system is not ready until the RS/6000 SP nodes reference the new IP address/hostname changes.

You will need to set the boot response to `customize` and then execute a `reboot` for the RS/6000 SP nodes. After the RS/6000 SP nodes have completed the reboot and customized, you should verify that the files on RS/6000 SP nodes reflect the new IP address/hostname changes.

The added Install support in AIX 4.1 for NIM requires the admin user to replace the NIM master objects. It may be necessary to update the NIM network and machine resources to match the new IP address or hostname. The NIM updates will be needed for the CWS and bootserver nodes.

If you had previously used System Partitioning on your RS/6000 SP system, it will be necessary to re-execute the system partition steps on your system. This will allow the necessary syspar config files to reflect any changes.

C.4 Updating the RS/6000 SP Node Interfaces

The following RS/6000 SP node updates need to be made on every RS/6000 SP node prior to updates made to the Control Workstation (CW).

1. Updating the RS/6000 SP node "`en0`" Interface

   You now need to update the "en0" interface on each RS/6000 SP node to allow the proper communication paths between the RS/6000 SP node and the Control Workstation. You need to change the en0 interface to the new IP address/hostname. You will also supply the proper gateway and route information for the en0 adapter. You do not need to modify any other adapter interfaces on the RS/6000 SP node. To accomplish this task you should use the `mktcpip` command. This can be executed through `smit mktcpip` or the `mktcpip` command on each RS/6000 SP node.

   The following are examples of using the `mktcpip` command.

   ```
   mktcpip -h <new node hostname> -a <new node IP addr> -m <subnet mask> \
   -i en0 -n <name server> -d <domain> -g <gateway IP addr> -t N/A
   ```

The following will change the hostname of node “k22n06” to be referenced as new node hostname “k88n06” in domain “ppd.pok.ibm.com.”

```
/usr/sbin/mktcpip -h'k88n06.ppd.pok.ibm.com' -a'129.40.88.70'      \
-m'255.255.255.192' -i'en0' -n'129.40.70.1' -d'ppd.pok.ibm.com'    \
-g'129.40.88.126' -t'N/A'
```

2. Update the NIM files and configuration

   a. You will need to update the /etc/niminfo file to provide the new and IP address and hostname changes for the nim client, master, and file references.

   b. If the node is a NIM master or a bootserver, you should remove all NIM objects from the node. This is accomplished by executing the following.

   ```
   /usr/sbin/nim -o unconfig master
   /usr/sbin/installp -u bos.sysmgt.nim.master
   ```

3. Update PSSP files for updates

   You will need to update the following files to place the new IP address and hostname information for node and server information.

   a. Update the /etc/SDR_dest_info file using new CW and SDR IP addresses

   b. Update the /etc/ssp/ files cw_name server_hostname, server_name to reference the new boot server hostname and IP address.

   c. Update the /etc/ssp/reliable_hostname file to reference the new node client hostname and IP address.

   d. Make sure that SP_NAME environment variable is updated or blank.

   e. Execute a /usr/lpp/ssp/kerberos/bin/kdestroy to remove any active kerberos ticket-granting-tickets. You may need to update the /etc/krb.conf file to point to the correct kerberos server.

Your update should now be completed at the RS/6000 SP node. You need to make the changes at the Control Workstation and then reboot the RS/6000 SP nodes.

C.5 Control Workstation IP Address/Hostname Changes

The following tasks need to be executed by the admin (root) user on the Control Workstation when changing the Ethernet IP addresses or hostname.

1. In a System Partition configuration, save the current SDR attributes by executing an archive /usr/lpp/ssp/bin/SDRArchive.

   You may want to backup the current /spdata filesystem in case there are problems doing the IP Address/hostname changes.

   You will need to reference the current and the new IP addresses/hostnames along with possible changes regarding system partition names (SP_NAME).

   You may want to reset the SP system back into one default system partition prior to changing the IP address hostname in the SDR.

2. Stop all the PSSP daemons sdr, hb, hr, hardmon on the CW.

   ```
   /bin/stopsrc -g sdr
   /bin/stopsrc -g hb
   /bin/stopsrc -g hr
   /bin/stopsrc -s hardmon
   ```

3. Remove the current source master objects for sdr, hb and hr for each system partition (SP_NAME) reference.

   ```
   /usr/lpp/ssp/bin/sdr -spname <SP_NAME> rmsrc
   /usr/lpp/ssp/bin/hb  -spname <SP_NAME> rmsrc
   /usr/lpp/ssp/bin/hr  -spname <SP_NAME> rmsrc
   ```

4. Using mktcpip or the chdev command, specify the new IP address or hostname changes required for the Control Workstation.

   If multiple system partitions exist, update the "/etc/rc.net" file to reference the new alias addresses used for system partitions. Execute the "/etc/rc.net" to configure the new alias addresses.

5. Manually update /etc/SDR_dest_info and /spdata/sys1/spmon/hmacls files to the new IP address and hostname used with hardmon and SDR.

   Move the system partition directories found at location /spdata/sys1/sdr/partitions from old IP addresses to the new IP addresses for each system partition reference.

   ```
   /bin/mv   <OLD IP_ADDR>   <New IP_ADDR>
   ```

6. Execute the setup_authent script to create authentication services for the new hostname being used. This step should only be executed for hostname changes for the CW. It is not required for changes made to IP addresses, or with RS/6000 SP node updates. Reference Step 12 in the RS/6000 SP Installation Guide. Manually check that the authentication files /etc/krb.conf, /etc/krb.realms, and /etc/krb-srvtab reference the new hostname.

7. You need to now create the new source master resources for the sdr, hb and hr daemons. This may be possible by executing the /usr/lpp/ssp/inst_root/ssp.basic.post_i script to recreate the daemons, and then start the daemons.

8. After the sdr daemon has been properly activated on the SP system, manually execute "SDRChangeAttrValues" for the Control Workstation.

   a. Change "hostname" attribute using new hostname for "SP_ports" object.

   b. Change "MACN" attribute using new hostname for "Frame" object.

   c. Change "control_workstation" attribute using short hostname for "SP" object.

9. You will want to remove the current Network Installation Manager (NIM) database and configuration files on the CW. You need to execute:

   ```
   /usr/sbin/nim -o unconfig master
   /usr/sbin/installp -u bos.sysmgt.nim.master
   ```

10. You may want to "reboot" the control workstation now. This will establish a clean system to reflect the IP address/hostname changes. You want to verify that all PSSP daemons are getting activated from the "/etc/inittab."

11. Using CMI or the spsitenv command, specify any hostname changes that may be referenced for ntp, printing, and user management. Reference Step 15 in the RS/6000 SP Installation Guide. (This is optional.)

12. Update the SDR objects using commands, or by using the SMIT-based 9076 SP Configuration Management Interface (CMI). You may need to execute the following activities from each system partition.

a. Using CMI or the spethernt command, specify the new Ethernet IP address or hostname change required for the RS/6000 SP nodes. Reference Step 18 in the RS/6000 SP Installation Guide.

b. Using CMI or the spadaptrs command, reset the “css0” and other adapters that need to reference the new Ethernet IP address/hostname that was changed. Reference step 22 in the RS/6000 SP Installation Guide.

c. Using CMI or the sphostnam command, reset the initial hostname that you want to use in your system. Reference Step 23 in the RS/6000 SP Installation Guide. (This is optional.)

d. Using CMI or the spbootins command, reset the boot install server to reference the new Ethernet IP address/hostname that was changed. You will need to set bootp response to “customize” for the RS/6000 SP nodes.

The spbootins will execute the setup_server command, which will create all the NIM-based files and resources required for installation. It will also create the authentication principals for the new RS/6000 SP nodes.

You may want to manually verify that the following files have been successfully updated after running the previously CMI steps:

```
/etc/ntp.conf
/tftpboot/<host>.config_info
/tftpboot/<host>.install_info
/tftpboot/<host>-new-srvtab
```

13. You may want to re-execute the system partitioning steps that are specified in Chapter 5 of the System Administrators Guide. Update other files on the CW that may reflect IP address or hostname changes.

    a. Update any files that are involved with hostname resolution. The files are /etc/hosts, /etc/resolv.conf (dns), and /var/yp/* (nis).

    b. Update the .rhosts or .klogin files that may exist.

    c. Update the /etc/filesystems and /etc/jmd_config.<SP_NAME> files for your RS/6000 SP configuration.

    d. You should also check to make sure your /tftpboot/script.cust file is updated to reflect IP address/hostname changes. Instead of hardcoding hostnames, you may reference the $SERVER and $CW variables.

14. You can now execute a “reboot” on each of the RS/6000 SP nodes. This can be accomplished using the System Monitor GUI. Follow the same install sequence as during installation: customize the boot servers first, and then customize the remaining RS/6000 SP nodes. For each bootserver node you will need to unconfigure NIM and re-execute setup_server to create the Installation files. Since the RS/6000 SP nodes are in customize mode, most configuration files will be updated to reflect the IP address/hostname changes.

Some files will not be updated on the nodes during the customization. You may want to “rcp” these files from the CW to the RS/6000 SP nodes by including them in the /tftpboot/script.cust file. These may include the /etc/resolv.conf, /.rhosts, and other RS/6000 SP customer-owned files.

C.6 Execution on RS/6000 SP Nodes

The following steps will need to be executed for IP address and hostname changes for both the Control Workstation and the RS/6000 SP nodes. These steps will be executed on each of the RS/6000 SP nodes.

1. After the reboot has completed, and your RS/6000 SP nodes have initialized, verify that the following have the correct IP address and hostname.
   - Updated SP Ethernet address specified for the RS/6000 SP node.
   - Updated default route is correct.
   - Hostname resolution files were updated correctly (/etc/hosts).
   - The /etc/SDR_dest_info file points to the CW and SDR addresses.
   - The acct_master hostname is updated for NFS export list directory /var/adm/acct (RS/6000 SP Accounting).
   - Files in directory /etc/ssp cw_name, server_hostname, server_name, and reliable_hostname point to the correct IP address/hostnames.
   - The files /etc/krb.conf, /etc/krb.realms, and /etc/krb-srvtab have the correct hostnames defined.
2. You may have to modify the following files on the RS/6000 SP nodes to reflect the updated IP addresses and hostname definitions.

   ```
   /etc/filesystems
   /etc/ntp.conf
   /etc/resolv.conf
   /.rhosts
   /.klogin
   ```

   If any of the files are not correct, make the proper updates for the correct IP address or hostname yourself.
3. You will want to remove the current Network Installation Manager (NIM) database and configuration files on any PSSP 2.1 bootserver node.

   ```
   /usr/sbin/nim -o unconfig master
   /usr/sbin/installp -u bos.sysmgt.nim.master
   ```

   You should then execute setup_server on the bootserver node.
4. It is best to reboot all of the RS/6000 SP nodes to reflect the IP address and hostname changes. When the RS/6000 SP nodes initialize, your RS/6000 SP system should be activated using the new IP addresses and hostnames.

Appendix D. Special Notices

This publication is intended to help technical professionals to better understand and be able to handle RS/6000 SP-related problems. The information in this publication is not intended as the specification of any programming interfaces that are provided by RS/6000 SP or POWERparallel System Support Programs. See the PUBLICATIONS section of the IBM Programming Announcement for RS/6000 SP and POWERparallel System Support Programs for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo
are trademarks or registered trademarks of Microsoft Corporation.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other trademarks are trademarks of their respective companies.

Appendix E. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 285.

- *PSSP 2.1 Technical Presentation*, SG24-4542
- *RS/6000 SP System Management: Easy, Lean, and Mean*, GG24-2563

E.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

E.3 Other Publications

These publications are also relevant as further information sources:

- *IBM RS/6000 SP Scalable POWERparallel Systems Administration Guide*, GC23-3897
- *IBM RS/6000 SP Scalable POWERparallel Systems Command and Technical Reference*, GC23-3900
- *IBM RS/6000 SP Scalable POWERparallel Systems Installation Guide*, GC23-3898
- *IBM RS/6000 SP Scalable POWERparallel Systems Diagnosis and Messages Guide*, GC23-3899
- *NIM Installation Guide*, SC23-2627

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number _____
- Credit card number _____

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

ACL	Access Control List
AIX	Advanced Interactive Executive
API	Application Programming Interface
BIS	Boot-Install Server
BSD	Berkeley Software Distribution
CPU	Central Processing Unit
CRC	Cyclic Redundant Code
CSS	Communication Subsystem
CWS	Control Workstation
EDC	Error Detection Code
EPROM	Erasable Programmable Read Only Memory
FIFO	First In - First Out
hb	heartbeat
HiPS	High Performance Switch
hrd	host respond daemon
HSD	Hashed Shared Disk
IBM	International Business Machines Corporation
IP	Internet Protocol
ISB	Intermediate Switch Board
ISC	Intermediate Switch Chip
ITSO	International Technical Support Organization
JFS	Journal File System
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitter Diode
LRU	Least Recently Used
LSC	Link Switch Chip
LVM	Logical Volume Manager
MIB	Management Information Base
MPI	Message Passing Interface
MPL	Message Passing Library
MPP	Massively Parallel Processors
NIM	Network Installation Manager
NSB	Node Switch Board
NSC	Node Switch Chip
OID	Object ID
ODM	Object Data Manager
PE	Parallel Environment
PID	Process ID
PROFS	Professional Office System
PSSP	Parallel System Support Program
PTC	Prepare to Commit
RAM	Random Access Memory
RCP	Remote Copy Protocol
RPQ	Request for Product Quotation
RVSD	Recoverable Virtual Shared Disk
SDR	System Data Repository
SMP	Symmetric Multiprocessors
SNMP	System Network Management Protocol
SRC	System Resource Controller
SSI	Single System Image
TCP/IP	Transmission Control Protocol / Internet Protocol
TOD	Time of Day
UDP	User Datagram Protocol
VSD	Virtual Shared Disk
VSM	Visual System Management

Index

Special Characters

A

B

C

D

E

N

P

R

S

T

U

V

W

This soft copy for use by IBM employees only.

Printed in U.S.A.