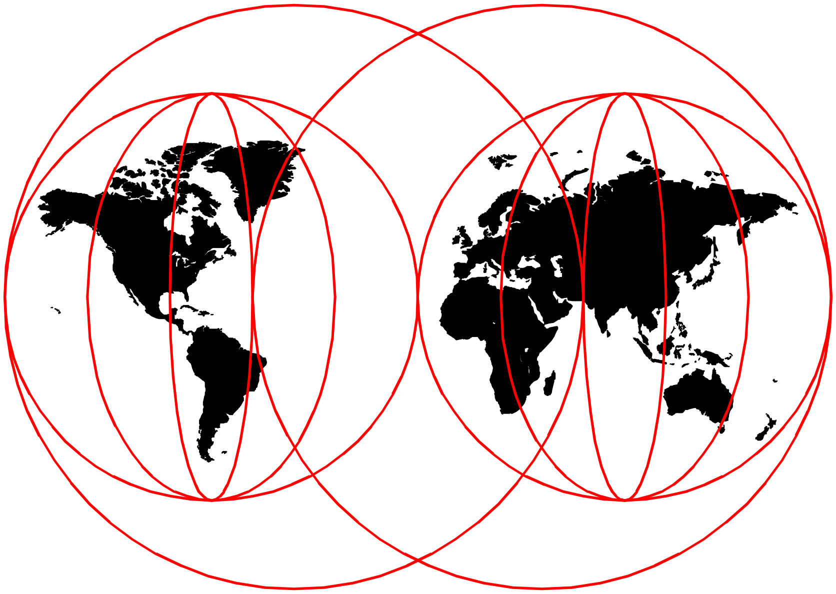
RS/6000 SP Software Maintenance

Hajo Kitzhöfer, Bärbel Altmann, Janakiraman Balasayee, Atul Sharma, Jorge Vergara



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5160-00

SG24-5160-00

International Technical Support Organization

RS/6000 SP Software Maintenance

July 1999

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 257.

First Edition (July 1999)

This edition applies to PSSP Version 2, Release 4 and Version 3, Release 1 of IBM Parallel System Support Programs for use with AIX 4.3.2

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

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

Contents

Figures

Tables

Preface

This redbook is intended for system administrators and system operators who need to manage an SP system. It will help you install, tailor and configure an RS/6000 SP system. It will also illustrate solutions for updating or migrating your SP to a higher level of AIX or PSSP software.

It is based on Version 2, Release 4 and Version 3, Release 1 of the POWERparallel System Support Program. The migration and update section covers procedures for going from PSSP 2.4 to PSSP 3.1. Nevertheless, most of the migration and update tasks described are general procedures that are more or less AIX and PSSP version-independent.

This redbook is also a good starting point for anyone wanting to get more background information about network install management (NIM), the switch commands and the use of Kerberos.

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.

Dr. Hajo Kitzhöfer is an Advisory International Technical Support Organization (ITSO) Specialist for RS/6000 SP at the Poughkeepsie Center. He holds a Ph.D. degree in Electrical Engineering from the Ruhr University of Bochum (RUB). Before joining ITSO, he worked as an SP Specialist at the RS/6000 and AIX Competence Center, IBM Germany. He has worked at IBM for eight years. His areas of expertise include RS/6000 SP, SMP, and Benchmarks. He now specializes in SP System Management, SP Performance Tuning and SP hardware.

Bärbel Altmann holds degrees in German Literature and Computer Science from Ludwig Maximilian University of Munich (LMU). Before she started working with AIX in 1990, she worked in the editorial section of a literary magazine published by LMU. While with IBM, she has worked on several major SP implementations at customer sites doing administration and problem determination, as well as consulting and second level support. Since 1997, she has worked as an SP instructor at the German Education and Training Center.

Janakiraman Balasayee is an IT Specialist in the PSS department of IBM Global Services, India. He holds a Diploma in Electrical Engineering. He has

14 years of experience in the IT industry. Before joining IBM, he worked on UNIX and Norsk Data Systems. He has worked at IBM India for three years. He provides country support for RS/6000, AIX and SP. His areas of expertise include RS/6000 and SP hardware, AIX and SP System Management.

Atul Sharma is an AIX Specialist in IBM U.K. He holds a diploma in Computer Studies from Thames Valley Polytechnic. He has five years of experience in AIX. Previously, he worked with retail systems and UNIX. Since joining IBM two years ago, he has worked in presales technical support. His areas of expertise include SP support and Capacity Planning.

Jorge Vergara is an AIX Specialist in IBM Colombia. He holds a degree in Systems Engineering from the Escuela Colombiana de Ingenieria. He has five years of experience in AIX. He works for the PSS division giving support mainly to SP systems and HACMP installations.

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

Rich Ferri
IBM PPS Lab Poughkeepsie

Linda Mellor
IBM PPS Lab Poughkeepsie

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 271 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an internet note to `redbook@us.ibm.com`

Chapter 1. Software Maintenance

This redbook concentrates on maintenance topics like installation, migration, update and backup of system software. It will help you install, tailor and configure an RS/6000 SP system. It also illustrates solutions for updating or migrating your SP to a higher level of AIX or PSSP software.

This publication is also a good starting point for anyone wanting to get more background information about network install management (NIM), the switch commands and the use of Kerberos.

First we clarify some terms so that we have a common understanding when we are using these terms in the following chapters.

1.1 Packaging Terms

AIX V4 is packaged into many different distinct groups of software. These groupings can be thought of as installable image groupings. Each grouping may have one or more images that may need to be installed in unique ways. The basic packaging concept is to install a set of packages that make up a base environment and then allow the users the option of installing additional packages as they wish. Optional software may be organized into *bundles*, which are groups of products that are suited for a particular use. For example, a client bundle provides client AIX functionality with minimum disk utilization.

This granularity allows customers to install exactly what they need to create their required environment, allowing a smaller minimum installation size. The packaging terms are explained in the following sections.

1.1.1 Licensed Program Products (LPPs)

These are complete software products, including all packages and filesets needed to provide a specific function. For example, the Base Operating System (BOS) is a Licensed Program Product (LPP).

1.1.2 Package

A *package* is a group of filesets with a common function collected into a single installable image. The image is in backup file format (BFF).

1.1.3 Filesets

AIX 4 is divided into *filesets*, each of which contains a group of logically related customer deliverable files. Each fileset can be individually installed and updated.

Revisions to filesets are tracked using the version, release, modification, and fix (VRMF) levels. Each time a fileset update is applied, the fix level is adjusted. Each time a maintenance level is applied, the modification level is adjusted, and the fix level is reset to zero.

Filesets provide a specific function. For example, bos.not.tcp.client is a fileset in the bos.net package (Base Network Support and Applications) providing TCP/IP client support.

A fileset is the smallest installable and updatable package in AIX. Customers can choose which functions to install by selecting filesets. Fixes are delivered to customers by providing an updated fileset, which contains only those files that have changed since that version/release of AIX was introduced.

1.1.3.1 VRMF

Revisions to filesets are tracked using a four-part Version.Release.Modification.Fix level. Along with each part of the VRMF versioning scheme come some implied rules for its use.

Version

A new version of AIX or a Licensed Program Product (LPP) implies a completely new design. In a new version, it is likely that every fileset will be changed. Although we strive for binary compatibility, some changes may be necessary that could break compatibility with older versions.

Release

A new release of AIX and LPP implies pervasive changes in the product or packaging. Most filesets are likely to be changed. Binary compatibility is usually preserved, although not guaranteed. Each release of an AIX version is supported independently, having its own service stream.

Modification

A modification level, also known as maintenance level, contains an accumulation of all fixes that were ever made to that release of the product, as well as fixes that were deferred. Maintenance levels are likely to contain support for new hardware, and may also contain minor functional enhancements. Binary compatibility with previous maintenance levels of the same release is guaranteed. Pervasive changes are discouraged, since they would likely have a profound impact on the size, instabilities, and risk associated with future individual fix packages.

Fix

A fix level is an update to a particular fileset. Each successive fix level contains all previous fixes to that fileset. The main goal for fix levels is to provide the smallest, most localized, lowest-risk fix possible to the customer. For this reason, any type of high-risk or pervasive change is discouraged.

1.1.4 Fileset Updates (PTFs)

Fileset updates, or Program Temporary Fixes (PTFs), are updates to particular filesets. A fileset update will either enhance, or correct a defect in, a previously installed fileset. Each successive fileset update contains all previous fixes to that fileset. Fileset updates are intended to provide the smallest, most localized, least-risk fix possible for a specific problem.

Figure 1 shows the relation between LPPs, optional program products and PTFs.

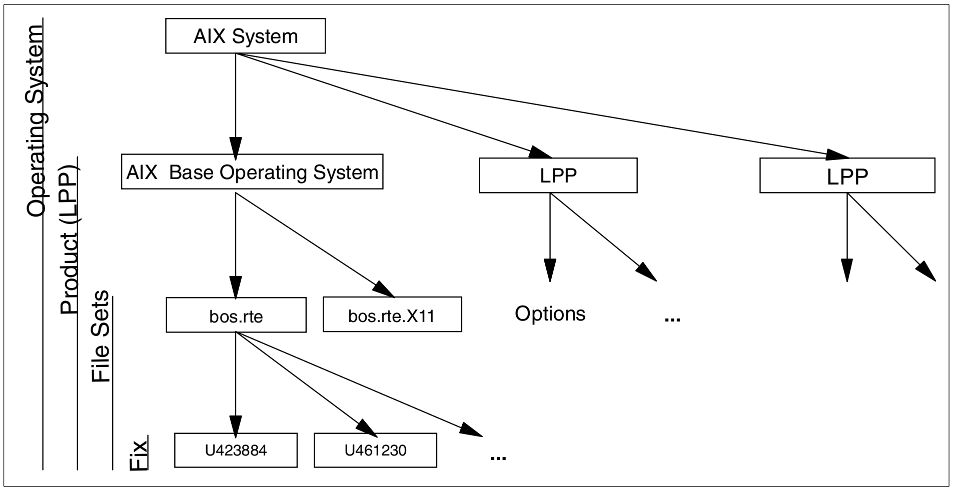


Figure 1. Architecture of LPPs, Optional Products and Fixes

1.1.5 Bundle

Bundles are collections of installable operating system software components and Licensed Program Product (LPP) components that are grouped together and can be installed with one selection. AIX V4.1 supports both system-defined and user-defined bundles.

The system-defined bundles are defined as:

- Client: A collection of software products for single-user systems running in a standalone or network client environment.
- Server: A collection of software products for multiuser systems running in a standalone or network environment.
- Personal Productivity: A collection of software products for graphical desktop systems running AIX and PC applications.
- Application Development: A collection of software products for developing applications, for example, compilers/linkers, libraries and debuggers.

1.1.6 Product Offerings

Product offerings are selected sets of packages which are sold together on physical media. Product offerings should not be confused with bundles (see 1.1.5, “Bundle” on page 4).

Examples of product offerings from IBM are:

- AIX 4.3 for Clients: This product offering is packaged with functionality aimed at satisfying the users of client systems. It is designed for customers who do not need to provide network server support for LAN/WAN-attached devices. These systems can be used as client systems, personal productivity workstations, print servers, name servers, and gateways.
- AIX 4.3 for Servers: This product offering is packaged and priced for full server-system functionality. These systems can be used as network file servers, data servers, print servers, iFOR/LS servers, compute servers, application servers, and so on.

Table 1 demonstrates the differences between LPP, package, and fileset.

Table 1. Example of Packaging Term Usage

LPP	Package	Fileset
bos	bos.INed	bos.INed
bos	bos.diag	bos.diag.rte
adacmp	adacmp	adacmp

1.2 AIX Maintenance Strategy

Fileset updates are available to customers on the World Wide Web using a Web browser, or can be downloaded using FixDist. FixDist is an AIX Motif client that handles downloading the requested fixes with all necessary requisites.

Customers that purchase Support Line can also request updates on media, either through the World Wide Web, or through the AIX Support Center.

1.2.1 Monthly Maintenance Packages

Customers sometimes want to order the latest available fixes. In practice, it would be a difficult task to find the latest fixes, and then order or download all of the fileset updates. About once per month, we make the latest available fileset updates for AIX easily orderable under a single APAR number.

1.2.2 Maintenance Levels

Preventive maintenance in AIX 4 is delivered in a maintenance level (ML). An ML consists of one fileset update for each fileset that has changed since the base level of the release. Each of these fileset updates is cumulative, containing all fixes for that fileset since the release was introduced, and supersedes all previous updates for the same fileset.

A new concept for preventive maintenance in AIX 4 is the recommended maintenance level (RML). A recommended maintenance level is a set of APARs since the most recent maintenance level.

Recommended maintenance levels are named using the four-digit VRMF level of the maintenance level with a two-digit number appended, starting with 01 and incrementing by 1 each time a new level is released.

Following are the maintenance levels and recommended maintenance levels produced for AIX 4.2:

ML 4210, also called AIX 4.2.1

RML 4210-01, also called AIX 4210-01

Following are the maintenance levels and recommended maintenance levels produced for AIX 4.3:

ML 4310, also called AIX 4.3.1

ML 4320, also called AIX 4.3.2

Maintenance levels for AIX 4 are released approximately twice a year while the release is active. Maintenance levels are likely to contain support for new hardware, and may also contain minor functional enhancements.

1.2.3 Maintenance-Level Bundle

A maintenance-level bundle is a collection of fixes and enhancements that update the operating system to the latest level. Software is selected for installation if it is in the bundle you choose and on the installation media.

1.3 Media - IBM Software Manufacturing Company

If a customer orders the AIX media option, media is delivered through IBM Software Manufacturing Company. Current media types are CD-ROM, 4mm, 8mm, and 1/4 inch tape.

1.3.1 CD-ROM

Due to increased reliability and speed, CD-ROM is by far the most popular media. The typical AIX order ships the following CD-ROMs:

1.3.1.1 AIX CDs

AIX 4.1 and 4.2 span two CD-ROMs. Both releases are currently available in client and server packages.

1.3.1.2 Bonus/Value Pack CD

AIX 4.1 includes a Value Pack CD, while AIX 4.2 includes a Bonus Pack CD. These CD-ROMs contain additional software such as Ultimedia Services and Netscape Navigator.

1.3.1.3 Update CD

The Update CD is a recent addition to the AIX CD-ROM set. This CD includes fixes for critical problems, optionally installable preventive maintenance, and may also contain new device support. Documentation for the update CD is included in both ASCII and HTML formats on the CD, with instructions for browsing the documentation on the CD jacket. The documentation includes descriptions of critical fixes, preventive maintenance, and device support, as well as installation instructions. This CD is updated approximately once per quarter.

1.3.2 Tape

The Stacked Product Option (SPO) tape not only contains AIX, but can also contain Licensed Program Products (LPPs) that the customer ordered at the same time as AIX.

In addition to AIX and LPPs, the SPO tape may also contain critical fixes. These fixes will be automatically installed along with the filesets they update. The optional preventive maintenance contained on the Update CD cannot be included on the tape media, since it too would automatically be installed, and would therefore no longer be optional.

Software Maintenance 8

Chapter 2. The Installation and Customization Process

One of the more complex tasks within an SP system is the installation of the nodes. In contradiction to standalone workstations which are usually equipped with tape- and/or CD-devices, all the SP nodes are missing these devices. Therefore the ordinary installation methods via tape or CD do not work. The solution for this dilemma is installation over a network. The administrative Ethernet, the connection between the CWS and all the nodes, is used for this installation method. That is one of the reasons why the Ethernet is mandatory.

Since version 3.2, AIX has had a network installation utility with very limited capabilities. The AIX 4 network installation management (NIM) is a totally new approach. NIM provides the ability to install machines with software from a centrally managed repository in the network.

Looking at the NIM installation methods, we notice that the PSSP software is only using a subset of the NIM capabilities (for example, only the installation of standalone workstations is used).

All of the NIM commands and options are being hidden by PSSP scripts and commands. Because of this, and as long as the installation is successful, some of the complexity appears to be transparent.

In order to diagnose installation problems, we need to understand what is actually happening under the covers. In this chapter we cover the various stages of the installation. We discuss how NIM is used in a usual AIX network environment and how PSSP utilizes these features during the installation phase.

2.1 NIM Overview

The NIM environment is comprised of client and server machines. A server provides resources (for example, files and programs required for installation) to another machine, the client. Any machine that receives resources is a client, although the same machine can also be a server in the overall network environment. Most installation tasks in the NIM environment are performed from one server, called the master.

The machines in an NIM environment, their resources, and the networks through which the machines communicate are all represented as objects within a central database that resides on the master. Each of these objects has attributes that give it a unique identity, such as the network address of a machine or the location of a file or directory. See Table 2 for an overview of NIM objects.

Table 2. Overview of NIM Objects

Object Class	Object Type	Object	Object Attributes
Machines	standalone dataless diskless	machines	TCP/IP hostname hardware address name of network type
Resources	spot lppsource mksysb scripts bosinst_data other resources	file directory	location
Networks	Ethernet token ring FDDI ATM	subnet	subnet mask

Not all the offered functionality of NIM is used in the SP. Only the minimum to install a standalone system is utilized. Table 3 shows the required information used during installation of an SP node.

Table 3. NIM Objects Used by PSSP

Object Class	**Object Type**	**Object**	**Object Attributes**
Machines	standalone	machines	TCP/IP hostname hardware address name of network type
Resources	spot lppsource mksysb scripts bosinst_data	file directory	location
Networks	Ethernet	subnet	subnet mask

The CWS is configured as a NIM master. The master and clients make up a NIM environment. The master provides resources to the clients. Referring to *AIX Version 4.3, Network Installation Management Guide and Reference*, SC23-4113, each NIM environment can have only one NIM master. As long as the CWS is the only boot/install server, this rule is not broken. As soon as you are using nodes as boot/install server as described in 2.1.1, “SP

Installation Hierarchy” on page 11, the rule is violated. Despite this fact, the hierarchical installation in the SP works.

As Figure 2 illustrates, NIM supports two ways of installing machines: pull or push.

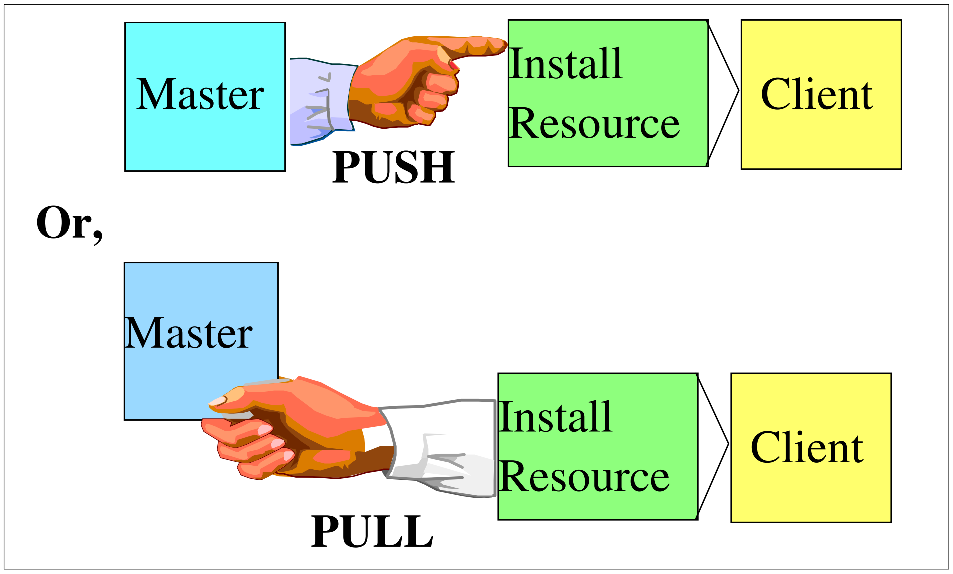


Figure 2. NIM Installation Methods

1. The push method is used when NIM installs dataless or diskless machines. This method is not used by the PSSP software.
2. The pull method is used when NIM installs standalone machines. Generally, pull means that the client invokes the installation methods. This is the mode used by the PSSP software.

2.1.1 SP Installation Hierarchy

As NIM installations utilize the network, the number of machines you can install simultaneously depends on the throughput of your network (namely, Administrative Ethernet). Other factors that can restrict the number of installations at a time are the disk access throughput of the installation servers, and the processor types of your servers.

To overcome this bottleneck you can configure nodes in your SP as boot/install server so that you will get a hierarchy of installation servers; Figure 3 on page 12 shows an example of a possible SP configuration.

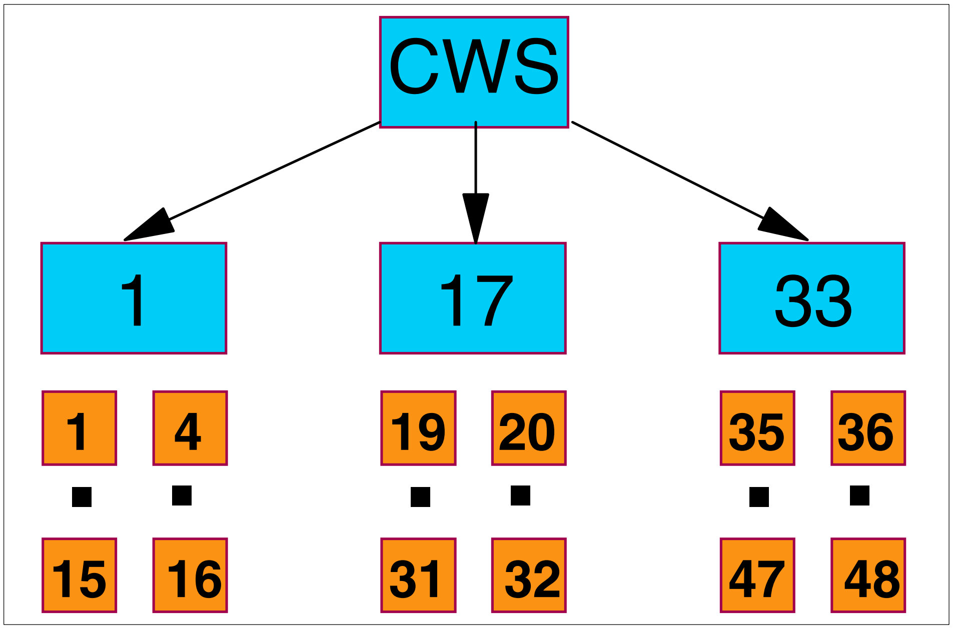


Figure 3. SP Installation Hierarchy

If your physical network structure supports such a hierarchy, you can initiate an installation process in each of these trees. In this case more nodes can be installed simultaneously than by using one flat network structure.

A NIM master makes use of the Network File System (NFS) utility to share resources with clients. As such, all resources required by clients must be local file systems on the master.

Administrators might be tempted to NFS-mount file systems from another machine to the CWS as /spdata/sys1/install/images to store node images, and as /spdata/sys1/install/<name>/lppsource to store AIX filesets due to disk space constraints. This is strictly not allowed, as NIM will not be able to NFS-export these file systems.

Figure 4 on page 13 shows what NIM looks like in an SP environment, and we describe how to configure this NIM environment.

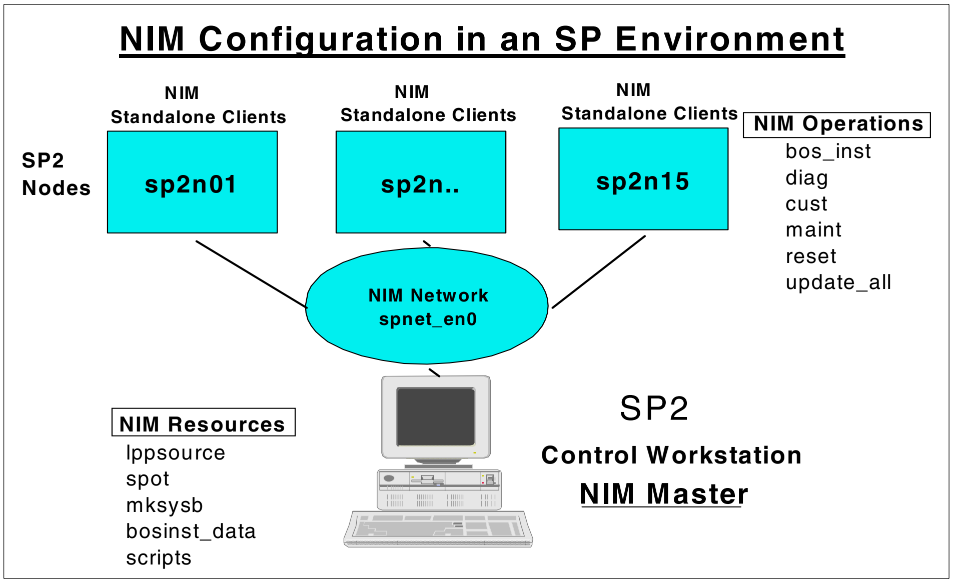


Figure 4. NIM Configuration in an SP Environment

The steps for creating a NIM master in an AIX environment to perform the various options are:

1. Create one of the standalone systems with AIX and NIM filesets installed as a NIM master, giving a unique network name for the primary network interface.
2. Define NIM standalone and diskless clients.
3. Create the basic installation resources like lpp_source and spot. If a customized mksysb image needs to be installed, then create mksysb resource. Create the bosinst_data resource if you would like to have a customized BOS install program.
4. Allocate the required resources for the clients based on the operations to be performed.
5. Invoke the operations you want to perform on the client machines.

In the following section we describe how to configure a general NIM environment on AIX. However, this section offers only a basic overview of NIM in AIX. For more information on NIM, see *AIX Version 4.3, Network Installation Management Guide and Reference*, SC23-4113*.*

Attention

In an SP environment, we never configure NIM directly (using SMIT, bottom line or WebSM). Instead, it is done using scripts called *wrappers*. Through this overview of NIM, it will be easier to know what is happening when you execute the wrappers.

2.1.2 NIM Master

The NIM master is fundamental to all operations in the NIM environment. It must be a machine running AIX with the NIM master fileset installed. The NIM master provides the central point of administration for the NIM environment and is the primary resource server. All other machines are clients to the master, including machines that may also serve resources. There is only one NIM master for each NIM environment.

The NIM filesets required to configure the NIM master are:

```
# lslpp -l |grep bos.sysmgt.nim
bos.sysmgt.nim.client      4.3.2.1  COMMITTED  Network Install Manager -
bos.sysmgt.nim.master      4.3.2.1  COMMITTED  Network Install Manager -
bos.sysmgt.nim.master_gui  4.3.2.0  COMMITTED  Network Install Manager - GUI
bos.sysmgt.nim.spot        4.3.2.1  COMMITTED  Network Install Manager - SPOT
bos.sysmgt.nim.client      4.3.2.0  COMMITTED  Network Install Manager -
```

The `rsh` command is used to remotely execute commands on clients. To use the `rsh` command, the $HOME/.rhosts file on the client is automatically configured by NIM when the client is initialized so that the master has the permissions required to run commands on the client as root. For more information on `rsh` and the .rhosts file, see *AIX Version 4.3, Network Installation Management Guide and Reference*, SC23-4113*.*

When you configure the NIM master, you specify a unique identifier to name the object that NIM creates to represent the network. This is the master's primary interface that connects to the clients. Once this object has been created, the name you specify identifies the network in all subsequent NIM operations.

In the SP CWS, the network type is created by default as spnet_en0 for the Ethernet network during installation. The SMIT fastpath to initialize the NIM master is `smitty nimconfig`.

To initialize the NIM master from SMIT, use the command:

```
# smitty nim
```

Select **Configure the NIM Environment.**

Select **Advanced Configuration**.

Select **Initialize the NIM Master Only.**

Enter the input fields as shown in the following screen:

```
 Configure Network Installation Management Master Fileset

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* Network Name                                         [spnet_en0]
* Primary Network Install Interface                    [en0]               +

  Allow Machines to Register Themselves as Clients?  [yes]               +
  Alternate Port Numbers for Network Communications
       (reserved values will be used if left blank)
    Client Registration                                []                   #
    Client Communications                              []                   #
```

2.1.3 NIM Networks

In order to perform certain NIM operations, the NIM master must be able to supply information necessary to configure client network interfaces. The NIM master must also verify that the clients are able to access all the resources of the master. When the NIM master is being configured, the network objects associated with the master are automatically defined in the NIM database.

You will have to define additional NIM networks if clients reside on other local area networks or subnets. In this case it is required to add a default or static NIM route between networks you specify. NIM routes are added to network definitions so NIM can determine the connectivity between NIM machines. When defining a default or static NIM route, you must also specify the gateways used by machines on the specified network.

The network types supported by NIM are tok, ent, fddi, generic and ATM. In the SP environment, the Ethernet (ent) network is always used as the primary network for NIM operations.

The SMIT fastpath to define the network is `smitty nim_mknet`.

2.1.4 NIM Machines

There are currently three types of machines that can be managed in the NIM environment. These are *standalone*, *diskless* and *dataless* clients. The NIM environment is composed of two basic machine roles: *master* and *client*.

The NIM master manages the installation of the rest of the machines in the NIM environment. The master is the only machine that can remotely run NIM commands on the clients. All the other machines in the NIM environment are clients to the master.

Standalone Clients

The standalone NIM clients are clients with the capability of booting and running from local resources. The standalone clients mount all the file systems from local disks and have a local boot image. They are not dependent upon network servers for their operation. They are managed in NIM networks primarily to install and update software.

Diskless and Dataless Clients

Diskless and dataless clients are machines that are not capable of booting and running without the assistance of servers on a network. The diskless clients have no hard disks and the dataless clients have hard disks, but they will not be able to hold all the data that is required for operating.

2.1.4.1 NIM Clients in the SP Environment

In an SP environment only standalone clients are supported. They have their own local disks to hold all the data that is required. In this section we get into more detail about standalone clients.

The fastpath to define a NIM standalone or diskless client on the NIM master is `smitty nim_mkmac`. Use the following to add the NIM standalone client sp4n10 using SMIT:

```
# smitty nim
```

Select **Perform NIM Administration Tasks.**

Select **Manage Machines.**

Select **Define a Machine.**

Host Name of Machine: `sp4n10`

Type of Network Attached to the Primary Network: `ent`

```
 Define a Machine

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* NIM Machine Name                                    [sp4n10]
* Machine Type                                        [standalone]            +
* Hardware Platform Type                              [rs6k]                  +
  Kernel to use for Network Boot                      [up]                    +
  Primary Network Install Interface
*   Cable Type                                         bnc                    +
*   NIM Network                                        spnet_en0
*   Host Name                                          sp4n10
    Network Adapter Hardware Address                  [10005AFA159D]
    Network Adapter Logical Device Name               [ent]
  IPL ROM Emulation Device                            []                      +/
  CPU Id                                              []
```

2.1.5 NIM Resources

Most operations on clients in the NIM environment require one or more resources. NIM resource objects represent files and directories that are used to manage the installation of standalone machines and the operation of diskless and dataless machines.

The NIM resources that are normally used for standalone clients in SP are:

- lppsource
- Shared Product Object Tree (SPOT)
- mksysb
- scripts
- bosinst_data

Let us look at what these resources are and how to create them using SMIT.

lpp_source

This is a directory containing all the installation images in bff format that can be installed on clients using the `installp` command. This directory should contain the minimum BOS filesets and all the device driver filesets so that the devices can be configured at the time of installation. Whenever you add or remove software from this directory, you must run the NIM *check* operation on the lpp_source to update the installation table-of-contents file for the resource.

In addition to updating the table-of-contents file for the lpp_source, the check operation also updates the *simages* attribute for the lpp_source, which indicates whether or not the lpp_source contains the images necessary to install the Base Operating images on the machine.

Check the simages attribute (simages=yes) after running the `nim -o check <lpp_source>` command.

To create the resource lppsource_aix432 in directory /spdata/sys1/install/aix432/lppsource using SMIT, use the following:

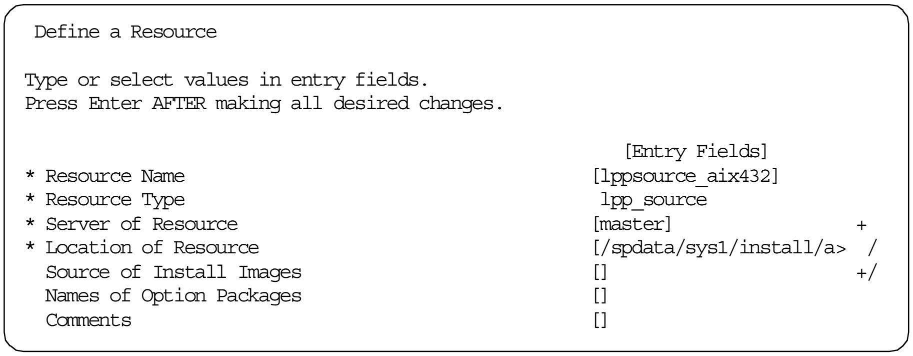
```
# smitty nim
```

Select **Perform NIM Administration Tasks.**

Select **Manage Resources.**

Select **Define a Resource.**

Resource Type: `lpp_source`



```
 Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Resource Name                                   [lppsource_aix432]
* Resource Type                                    lpp_source
* Server of Resource                              [master]              +
* Location of Resource                            [/spdata/sys1/install/a>  /
  Source of Install Images                        []                    +/
  Names of Option Packages                        []
  Comments                                        []
```

SPOT

A Shared Product Object Tree (SPOT) provides a /usr file system for diskless and dataless clients, as well as the network boot support for all clients. This is the fundamental resource in an NIM environment. It is required to install or initialize all machine configuration types.

Everything that a machine requires in a /usr file system, such as the AIX kernel, executable commands, libraries, and applications are included in the SPOT. Machine-unique information or user data is usually stored in the other file systems. A SPOT can be located on any standalone machine within the NIM environment, including the master. The SPOT is created, controlled, and maintained from the master, even though the SPOT can be located on another system.

There are two ways to create a SPOT. You can convert the /usr file system (/usr SPOT), or you can locate the SPOT elsewhere within the file system (non-/usr SPOT) on the server.

The PSSP software uses the non-/usr SPOT method for the NIM setup. All SPOTs are located under /spdata/sys1/install/<aix version>. Here is an example where the POT for AIX 4.3.2 will be created:

/spdata/sys1/install/aix432/spot

The /usr SPOT inherits all the optional software that is already installed on the server. All the clients using the /usr SPOT have access to the optional software installed on the server. The non-/usr SPOT can be used to manage a different group of optional software than that installed and licensed for the server.

Creating a SPOT by converting the /usr file system has the advantage of being fast and using much less disk space. However, this method does not give you the flexibility to choose which software packages will be included in the SPOT, because all the packages and filesets installed in the /usr file system of the machine serving the SPOT will be included in the SPOT. This was the way PSSP 2.2 was using the SPOT, which led to some problems.

Starting with PSSP 2.2, the second method, creating a non-/usr SPOT, is used. This method uses a lot more disk space, but it is more flexible. Initially, only the minimum set of software packages required to support NIM clients is installed in the SPOT, but additional packages and filesets can be installed. Also, it is possible to have multiple SPOTs, all with different additional packages and filesets installed, serving different clients.

That is the way the PSSP software supports different AIX versions on SP nodes. For each group of nodes with the same AIX version, a common SPOT is maintained.

A SPOT varies in size from 100 MB up to, and sometimes in excess of, 300 MB, depending on the software that is installed. Since all device support is installed in the SPOT and the number of device filesets typically increases, the size is not easily predictable from release to release.

SPOTs are used to support all NIM operations that require a machine to boot over the network. These operations are as follows:

- bos_inst
- maint_boot
- diag
- dkls_init
- dtls_init

The last two options are not used in an SP environment.

When a SPOT is created, network boot images are constructed in the /tftpboot directory of the SPOT server, using code from the newly created SPOT. When a client performs a network boot, it uses tftp to obtain a boot image from the server. After the boot image is loaded into memory at the client, the SPOT is mounted in the client's RAM file system to provide all additional software support required to complete the operation.

Each boot image created is up to 4 MB in size. Before creating a SPOT, ensure there is sufficient space in the root (/) file system, or create a separate file system for /tftpboot to manage the space required for the network boot images.

Recommendation

To avoid running out of filesystem space in the root filesystem we strongly recommend creating a separate file system, /tftpboot. The size of this filesystem should be at least 60 MB.

The Micro Channel-based systems support booting from the network using Token-Ring, Ethernet, or FDDI. The PowerPC PCI bus-based systems support booting from the network using Token-Ring or Ethernet. The uniprocessor MCA and PCI bus-based systems can be used in a diskless or dataless configuration. Again, the diskless and dataless options are not used in an SP environment.

A single network boot image can be accessed by multiple clients; therefore, the network boot image cannot contain any client-specific configuration information. The platform type is specified when the machine object is defined, while the network type is determined from the primary interface definition. Two files are created in the /tftpboot directory on the SPOT server for each client to be network-booted: ClientHostName and ClientHostName.info. The ClientHostName file is a link to the correct network boot image, while the ClientHostName.info file contains the client configuration information.

When the SPOT is defined (and created), the following occurs:

The BOS image is retrieved from archive or, for /usr conversion, just the root directory is retrieved from archive (/usr/lpp/bos/inst_root). The device support required to support NIM operations is installed. Network boot images are created in the /tftpboot directory.

Each network boot image supports a single network, platform, and kernel type. The network boot image files are named SPOTName.Platform.Kernel.Network. The network types are Token-Ring, Ethernet, and FDDI. The platform types are:

rs6k

used for POWER/POWER2/P2SC/PowerPC MCA bus-based machines

rspc

used for PowerPC Reference Platform (PREP) Architecture-based machines

chrp

used for PowerPC Common Hardware Reference Platform (CHRP) Architecture-based machines

The rs6ksmp platform for 4.2 (and later) SPOTs are represented by the boot image with a platform type of rs6k and a kernel type of mp.

The kernel types are:

up

used for single processor machines

mp

used for multiple processor machines

Both up and mp boot images are created for each platform and network type. The network boot images located in /tftpboot look similar to the following:

spot_aix432.rs6k.mp.ent

spot_aix432.rs6k.mp.fddi

spot_aix432.rs6k.mp.tok

spot_aix432.rs6k.up.ent

spot_aix432.rs6k.up.fddi

spot_aix432.rs6k.up.tok

spot_aix432.rspc.mp.ent

spot_aix432.rspc.mp.tok

spot_aix432.rspc.up.ent

spot_aix432.rspc.up.tok

The amount of space used in the /tftpboot directory for boot images may become very large. A SPOT that supports network boot for all possible combinations of platforms, kernel types, and network adapters may require as much as 60 MB in /tftpboot. If the same server serves multiple SPOTs, the space required in /tftpboot will be even more since each SPOT creates its own set of boot images.

In AIX 4.3, NIM creates by default only the boot images required to support the machines and network types that are defined in the environment. This should significantly reduce the amount of disk space used and the time required to create boot images from SPOT resources.

A SPOT will be created by using the following SMIT shortcut:

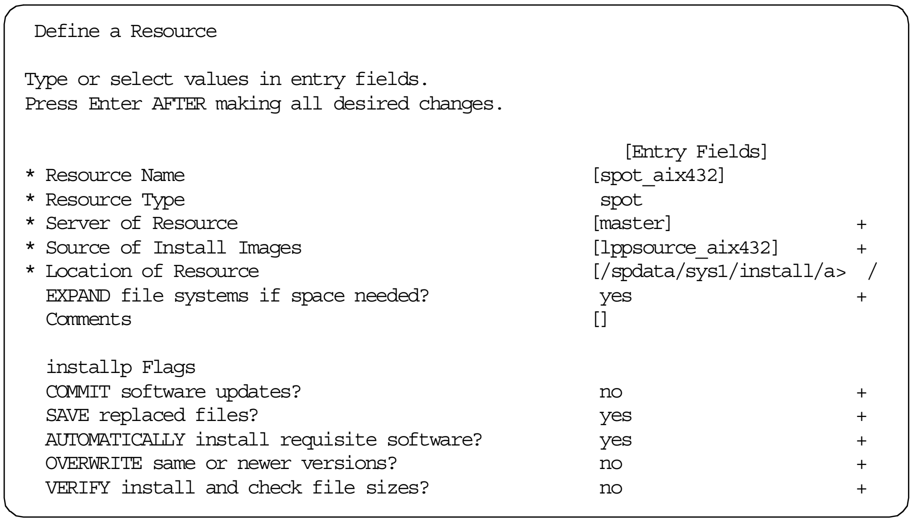
```
# smitty nim
```

Select **Perform NIM Administration Tasks**.

Select **Manage Resources.**

Select **Define a Resource.**

Resource Type: `spot`



```
 Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
* Resource Name                                   [spot_aix432]
* Resource Type                                    spot
* Server of Resource                              [master]              +
* Source of Install Images                        [lppsource_aix432]    +
* Location of Resource                            [/spdata/sys1/install/a>  /
  EXPAND file systems if space needed?             yes                  +
  Comments                                        []

  installp Flags
  COMMIT software updates?                         no                   +
  SAVE replaced files?                             yes                  +
  AUTOMATICALLY install requisite software?        yes                  +
  OVERWRITE same or newer versions?                no                   +
  VERIFY install and check file sizes?             no                   +
```

mksysb

An mksysb resource represents a file that is a system backup image created using the `mksysb` command. The type of resource can be used as the source for the installation of a client. The file must reside on the hard disk of the NIM master in order to be defined as a resource. It cannot be located on a tape or other external media (like CDs).

First create the mksysb image on a file from a configured system; never try to use the mksysb image of a system with NIM master initialized.

In this example, we copied the mksysb image file as bos.obj.ssp.432 to directory /spdata/sys1/install/images before going to SMIT to configure the mksysb resource named mksysb_1.

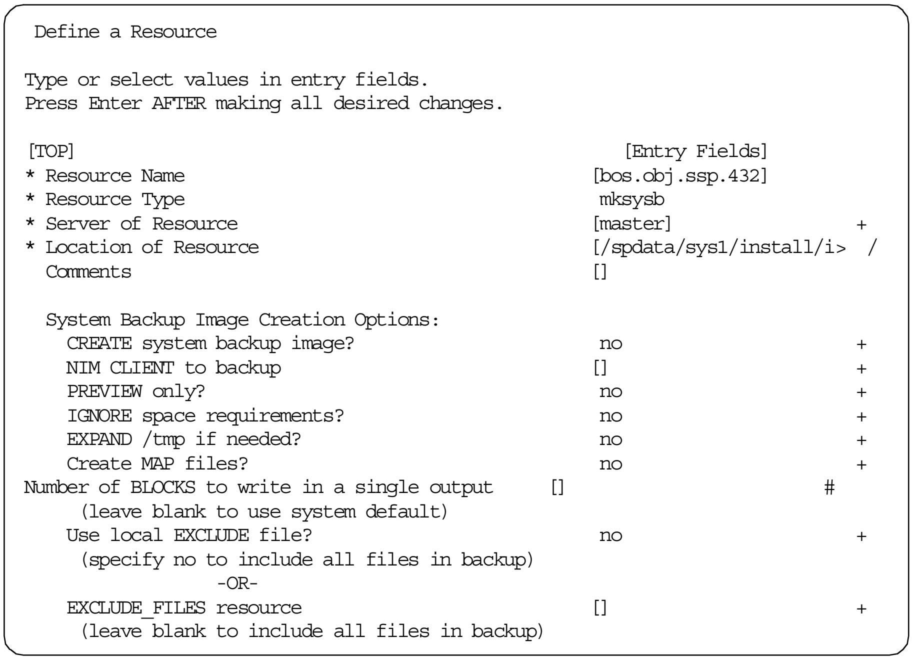
```
# smitty nim
```

Select **Perform NIM Administration Tasks.**

Select **Manage Resources.**

Select **Define a Resource.**

Resource Type: `mksysb`



```
 Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                       [Entry Fields]
* Resource Name                                          [bos.obj.ssp.432]
* Resource Type                                           mksysb
* Server of Resource                                     [master]              +
* Location of Resource                                   [/spdata/sys1/install/i>  /
  Comments                                               []

  System Backup Image Creation Options:
    CREATE system backup image?                           no                   +
    NIM CLIENT to backup                                 []                    +
    PREVIEW only?                                         no                   +
    IGNORE space requirements?                            no                   +
    EXPAND /tmp if needed?                                no                   +
    Create MAP files?                                     no                   +
Number of BLOCKS to write in a single output     []                        #
     (leave blank to use system default)
    Use local EXCLUDE file?                               no                   +
     (specify no to include all files in backup)
                  -OR-
    EXCLUDE_FILES resource                               []                    +
     (leave blank to include all files in backup)
```

bosinst_data

This resource customizes the BOS install flow (it automates menu choices). It is used to customize the installation flow and to avoid the need for prompting at the console. To create this resource, first copy the file /usr/lpp/bosinst.template as bosinst_data to the /spdata/sys1/install/pssp directory and edit the file to customize your environment.

The contents of the bosinst_data file for a no-prompt installation are shown in the following screen:

```
control_flow:
    BOSINST_DEBUG = yes
    CONSOLE = /dev/tty0
    INSTALL_METHOD = overwrite
    PROMPT = no
    EXISTING_SYSTEM_OVERWRITE = yes
    INSTALL_X_IF_ADAPTER = no
    RUN_STARTUP = no
    RM_INST_ROOTS = no
    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
```

After editing this file, we can create the no-prompt resource using SMIT as follows:

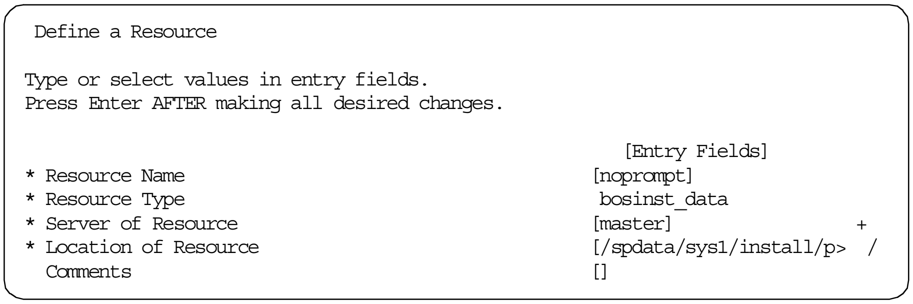
```
# smitty nim
```

Select **Perform NIM Administration Tasks.**

Select **Manage Resources.**

Select **Define a Resource.**

Resource Type: `bosinst_data`



```
 Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* Resource Name                                       [noprompt]
* Resource Type                                        bosinst_data
* Server of Resource                                  [master]              +
* Location of Resource                                [/spdata/sys1/install/p>  /
  Comments                                            []
```

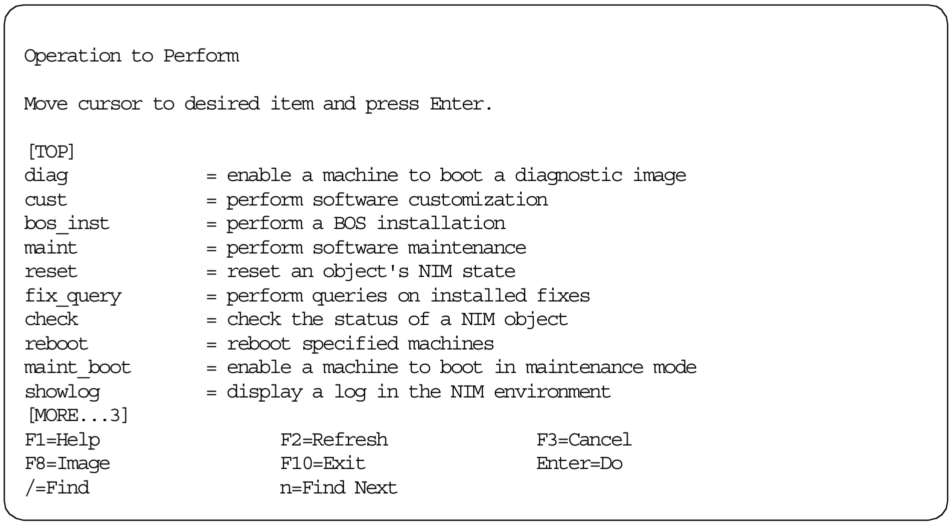
script

This is a customization program executed after installation. It is only a shell script which you create with any filename and define as a script resource for getting executed after the installation of the mksysb image or the SPOT image. This script can be used for customizing your environment in the client system, such as increasing the paging space and the file system size, configuring the NIS client, and so on. The steps for configuring this resource are the same as for creating the bosinst_data resource.

Note: You do not have a full AIX environment during the NIM customization process. Also, default routes and additional adapters are not configured at this stage of the installation.

2.1.6 NIM Operations on Standalone Clients

Although an installed client is capable of booting from the local disk, it may be necessary to perform a network boot of a client for certain NIM operations.



```
Operation to Perform

Move cursor to desired item and press Enter.

 [TOP]
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
fix_query        = perform queries on installed fixes
check            = check the status of a NIM object
reboot           = reboot specified machines
maint_boot       = enable a machine to boot in maintenance mode
showlog          = display a log in the NIM environment
 [MORE...3]
F1=Help                 F2=Refresh              F3=Cancel
F8=Image                F10=Exit                Enter=Do
/=Find                  n=Find Next
```

The operations that can be performed on the standalone clients are as follows:

bos_inst

This operation is used to install the AIX Base Operating System on standalone clients. The standalone clients are the boot/install servers and the nodes.

diag

This is used to boot the clients into diagnostics mode to perform hardware maintenance.

cust

This is used to install software filesets and updates on standalone clients and SPOT resources.

fix_query

This is used to display whether specified fixes are installed on a client machine or SPOT resources.

lppchk

This is used to verify that software was installed successfully by running the `lppchk` command on a NIM client or SPOT resource.

maint

This is used to deinstall software filesets and commit and reject updates on standalone clients and SPOT resources.

maint_boot

This is used to prepare resources for a client to be network-booted into maintenance mode to perform software maintenance.

reset

This is used to change the state of a NIM client or resource, so NIM operations can be performed with it. A reset may be required on a machine or resource if an operation was stopped before it completed successfully.

check

This is used to verify the usability of a machine or resource in the NIM environment. The check operation can be performed on NIM clients, or a group of NIM clients, a SPOT resource, or an lpp_source resource.

showlog

This is used to list software installed on a NIM client or SPOT resource.

reboot

This is used to reboot a NIM client machine. The target of a reboot operation can be any standalone NIM client or groups of clients.

The fastpath to perform a NIM operation on a standalone or diskless/dataless client is `smitty nim_mac_op`.

The following shows the steps to perform a bos_inst operation on node sp4n10 using the mksysb_1 image:

1. First allocate the resources for the client sp4n10 to get access to the files in the master. In addition to the mksysb_1 resource, allocate the lppsource and the SPOT resource for the node to do a network boot. To allocate the resources to the client sp4n10 using SMIT, use the following:

```
# smitty nim
```

Select **Perform NIM Administration Tasks.**

Select **Manage Machines.**

Select **Manage Network Install Resource Allocation.**

Select **Allocate Network Install Resources.**

Target name: `sp4n10`

Available Network Install Resources: (select the following)

`noprompt`

`lppsource_aix432`

`spot_aix432`

`mksyb_1`

`psspscript` (the customization script)

2. The next step is to initiate the bosinst operation on the client sp4n10. To perform this using SMIT, use the following:

```
# smitty nim
```

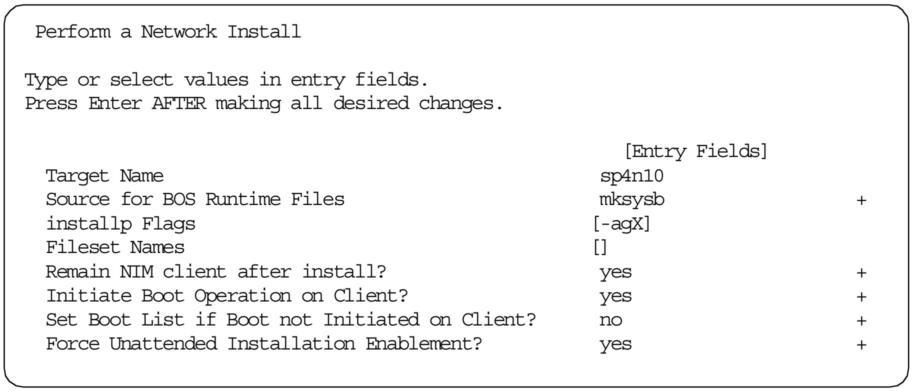
Select **Perform NIM Administration Tasks.**

Select **Manage Machines.**

Select **Perform Operations on Machines.**

Target Name: `sp4n10`

Operation to perform: `bos_inst`

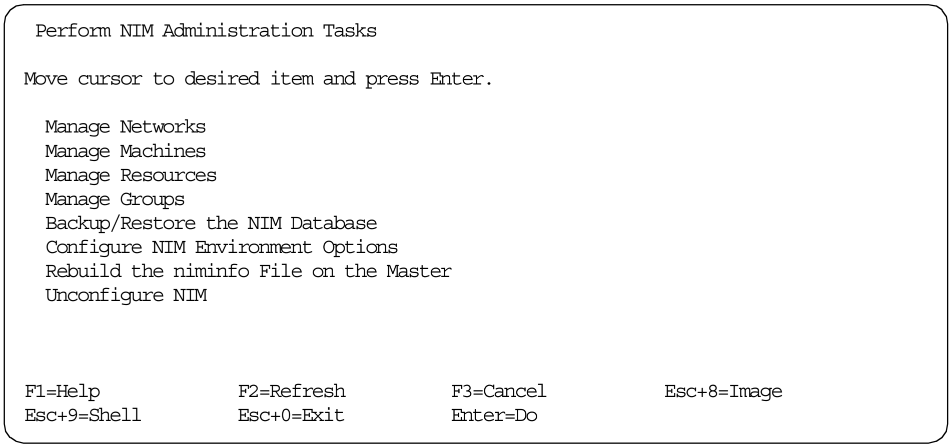


```
 Perform a Network Install

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Target Name                                         sp4n10
  Source for BOS Runtime Files                        mksysb              +
  installp Flags                                     [-agX]
  Fileset Names                                      []
  Remain NIM client after install?                    yes                 +
  Initiate Boot Operation on Client?                  yes                 +
  Set Boot List if Boot not Initiated on Client?      no                  +
  Force Unattended Installation Enablement?           yes                 +
```

2.1.7 NIM Administration

Generally the problems encountered during SP installation are related to NIM. To resolve NIM-related problems, always use the following SMIT menu. From this SMIT screen you can administer the clients and resources, and also check the state of the various NIM objects.

The fast path to reach this menu is `smitty nim_admin`.



```
 Perform NIM Administration Tasks

Move cursor to desired item and press Enter.

  Manage Networks
  Manage Machines
  Manage Resources
  Manage Groups
  Backup/Restore the NIM Database
  Configure NIM Environment Options
  Rebuild the niminfo File on the Master
  Unconfigure NIM



F1=Help             F2=Refresh          F3=Cancel           Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Note: In many cases, NIM prevents operations on a target object when the object is not in a ready state. In such situations you can either try to reset the object to a ready state or set the force option for an operation to "on" when the action is initiated. This will ignore the state of the target object and the operation can be performed on it. In NIM the force option is available for the following operations:

Manage Networks

This displays a menu of operations that enable you to manage NIM networks. Operations include creating a network, changing or showing the characteristics of a network, removing a network, and managing routing information for a network.

Note: You must have root user authority to perform any of the network operations.

Manage Machines

This displays a menu of operations that can be performed on NIM machines. The machine operations include adding a machine, changing or viewing the characteristics of a machine, and removing a machine. Additional machine operations are managing network interfaces, managing resource allocation, and performing machine control actions.

Manage Resources

This displays a menu of operations that enable you to manage NIM resources. The operations include listing the existing resources, adding a resource, changing or showing the characteristics of a resource, removing a resource, and performing maintenance on a Shared Product Object Tree (SPOT).

In an existing customized NIM environment, when there is a problem related to one of the resources, you can check the status of that resource from this menu. If you find a problem, it is better to reset that resource with the force option set to `yes`. This will reset the resource object and bring it to a ready state to perform a NIM operation.

Manage Groups

This displays a menu of operations that can be performed on NIM machine groups. Machine groups provide a means of grouping related NIM machines so that the machines may be operated upon as a logical unit. This will be useful when you want to perform similar operations in multiple clients.

Backup/Restore of the NIM Database

We recommend that you take a backup of the customized NIM database, so that you can restore it in case of a problem. The steps for taking a NIM database backup are documented in 7.1.4, “Backing Up the NIM Database” on page 146. The steps for restoring are documented in 7.2.4, “Restoring the NIM Database” on page 152.

2.2 NIM Configuration Using PSSP

In an SP environment, we never configure NIM manually. Instead, the PSSP software package has scripts called *wrappers* for configuring the NIM environment. Wrappers are functions written in Perl and each one is an independent script.

These wrappers are called by the `setup_server` command for configuring the NIM environment in the CWS and Boot/Install servers (BIS). These scripts can also be called from the command line if you know the sequence in which they have to be executed. These scripts use the information from the SDR to initiate the appropriate NIM commands with the required options. In this section we discuss how NIM is configured in an SP using these wrappers.

The NIM wrappers are part of the ssp.basic filesets. They are:

- mknimmast
- mknimint
- mknimclient
- mkconfig
- mkinstall
- mknimres
- delnimmast
- delnimclient
- allnimres
- unallnimres

Here we discuss the basic overview of these wrappers. For more information on the logic flow within each script, refer to Section 2.2.2 of *RS/6000 SP: PSSP 2.2 Survival Guide,* SG24-4928.

2.2.1 mknimmast

This wrapper initializes the NIM master. This is the first step for configuring a NIM master. In order to configure the NIM master, the bos.sysmgt.nim.master and bos.sysmgt.nim.spot filesets need to be installed. This wrapper is executed only on the CWS and BIS as part of the `setup_server` script using information from the SDR to configure the NIM master.

Command syntax:

```
# mknimmast -l <node_number_list>
```

`-l<node_number_list>` is the list of node numbers to define as NIM masters. This is a required option. Node number 0 signifies the CWS.

Example:

```
# mknimmast -l 0
```

This command initializes the CWS as the NIM master.

2.2.2 mknimint

This wrapper creates network objects on the NIM master to serve the NIM clients. If there is more than one NIM master configured, `mknimint` creates network objects to find the CWS. In cases where more than one NIM master is defined, dedicated nodes become Boot/Install servers (BISs). The CWS remains as a resource center for other NIM masters. Therefore, it is important to configure network paths for reaching the CWS. This is the reason why the BIS, while executing `mknimint`, searches for network interfaces of the CWS using the `netstat` command. To make sure that the BIS can reach every network interface of the CWS, the route definitions must be in place.

Command syntax:

```
# mknimint -l <node_number_list>
```

`-l <node_number_list>` is the list of node numbers to define as NIM masters. This is a required option. Node number 0 signifies the CWS.

Example:

```
# mknimint -l 0
```

This command configures the NIM network objects for all the interfaces in the CWS.

2.2.3 mknimclient

This wrapper takes input from the SDR to create the clients on the CWS and the BIS that are configured as NIM masters. The SP node's reliable hostname configured in the SDR is used as the NIM machine object name.

Command syntax:

```
# mknimclient -l <node_number_list>
```

`-l <node_number_list>` is the list of node numbers to define as NIM clients. This is a required option. Node number 0 (the CWS) is not allowed.

Example:

```
# mknimclient -l 5
```

This command creates the NIM client definitions for node 5 on the NIM master.

2.2.4 mkconfig

This wrapper creates the /tftpboot/<reliable_hostname>.config_info file for every node. The input values for this script are retrieved from the SDR for all the nodes. This command has no options, and whenever it is executed it creates the config files for all nodes that have a bootp_response value not set to *disk*. This file is used during network installation of the nodes.

Command syntax:

```
# mkconfig
```

2.2.4.1 The config_info File

The config_info file is used by the pssp_script. It exports all the required environment variables during the installation phase. You can see the contents of a sample config_info file in the following screen.

```
# cat /tftpboot/sp4n09.install_info

#!/bin/ksh
export control_workstation="9.12.0.4 192.168.4.140"
export cw_hostaddr="192.168.4.140"
export cw_hostname="sp4en0"
export server_addr="192.168.4.140"
export server_hostname="sp4en0"
export rel_addr="192.168.4.9"
export rel_hostname="sp4n09"
export initial_hostname="sp4n09"
export auth_ifs="sp4en0/192.168.4.140"
export authent_server="ssp"
export netinst_boot_disk="hdisk0"
export netinst_bosobj="bos.obj.ssp.432"
export remove_image="false"
export sysman="true"
export code_version="PSSP-3.1"
export proctype="UP"
export LPPsource_name="aix432"
export cwsk4=""
export LPPsource_hostname="sp4en0"
export LPPsource_addr="192.168.4.140"
export platform="rs6k"
export ssp_jm="yes"
```

2.2.5 mkinstall

This wrapper creates the /tftpboot/<reliable_hostname>.install_info file for every node in the SDR whose bootp_response is not set to *disk*. This file is used during the network installation of the nodes.

Command syntax:

```
# mkinstall
```

2.2.5.1 The install_info File

The install_info file is used by the pssp_script. The information from this file is taken for the base configuration of the node. See an example file in the following screen.

```
cat /tftpboot/sp4n09.config_info

9 sp4n09.msc.itso.ibm.com 192.168.4.9 192.168.4.140 8 1 4 3 1 yes rootvg true hd
isk0 1 false
en0 192.168.4.9 255.255.255.0 NA 10 "" 192.168.4.0
css0 192.168.14.9 255.255.255.0 NA NA "" 192.168.14.0
```

The fields of the config_info file contain the following:

- Node number
- Initial hostname
- Node IP address
- Switch node number
- Switch number (switch board)
- Switch chip port
- Slot that this node use
- If arp is enabled for the switch

This additional information is only for PSSP >= 3.1 clients:

- Selected volume group
- Quorum
- Physical volume list
- Number of copies (mirroring)
- Volume group mapping

An entry for each adapter is defined in the SDR, listing:

- Adapter name
- Adapter IP address
- netmask
- ring_speed (for Token Ring)
- bnc_select (dix/bnc/tp for Ethernet)

This additional information is only for PSSP >= 3.1 clients:

- Ethernet rate
- Duplex type

2.2.6 mknimres

This wrapper creates all the NIM resources for installation, diagnostics, migration and customization. The resources created will be used by the allnimres wrapper for allocation to the clients, depending on the bootp_response field.

Command syntax:

```
# mknimres -l <node_number_list>
```

`-l <node_number_list>` is the list of node numbers to define as NIM masters. This is a required option. Node number 0 signifies the CWS.

Example:

```
# mknimres -l 1
```

This command creates all the resources in the CWS.

The resources that are created by this wrapper are shown in the following screen:

```
psspscript          resources       script
prompt              resources       bosinst_data
noprompt            resources       bosinst_data
migrate             resources       bosinst_data
lppsource_aix432    resources       lpp_source
mksysb_1            resources       mksysb
spot_aix432         resources       spot
```

The purpose and location of these resources are as follows:

psspscript:

The pssp_script file is copied from the /usr/lpp/ssp/install/bin/pssp_script to the /spdata/sys1/install/pssp directory when the `mknimres` command is executed. This pssp_script is called by NIM after installation of a node and before NIM reboots the node.

It is run under a single user environment with the RAM file system in place. It installs the required LPPs and does the post-installation setup.

This script takes the input from the */tftpboot/<node>.config_info* and */tftpboot/<node>.install_info* files.

prompt:

This resource is allocated when you want the node to prompt for the input from the console. It is used to perform a maintenance or diagnostic mode of operation.

noprompt:

This resource is allocated when installing or migrating the node using full overwrite install and you do not want the installation to prompt for any input on the console.

migrate:

This resource is allocated when you want to perform a migration on the nodes to a newer version of AIX.

lppsource_aix432:

This resource contains the BOS minimum filesets and PTFs required for installing the nodes in the installp format. Whenever you copy any new filesets or PTFs to this directory, you must initialize the table of contents file (.toc file) and also update the SPOT resource to reflect these PTFs.

mksysb_1:

This resource contains the image to be installed on the nodes. Along with the SP system, you get a spimg tape that contains the minimum BOS filesets and PTFs in image file format. If you are not using this minimal image and want to use the image you have created, then make sure that all the minimum PTFs are also installed. This image is copied to the directory /spdata/sys1/install/images as part of the installation steps.

spot_aix432:

This resource is defined as a directory structure that contains the runtime files common to all machines. It is created under the directory /spdata/sys1/install/<aix version>/spot.

2.2.7 delnimmast

This wrapper is used to delete the NIM master definitions and all the NIM objects on the CWS or boot/install server. The NIM filesets will also be deinstalled by this script.

Command syntax:

```
# delnimmast -l <node_number_list>
```

`-l <node_number_list>` is the list of node numbers to undefine as NIM masters. This is a required option. Node number 0 signifies the CWS.

Example:

```
# delnimmast -l 0
```

This command can be used to unconfigure the NIM master definitions in the CWS.

2.2.8 delnimclient

This wrapper is used to deallocate the resources and to delete the NIM client definitions from the NIM master.

Command syntax:

```
# delnimclient -l <node_number_list> | -s <server_node_list>
```

`-l <node_number_list>` is the list of node numbers which will be deleted as NIM clients. This is a required option. Node number 0 (the CWS) is not allowed.

`-s <server_node_list>` is the list of server (NIM master) nodes on which to delete all clients that are no longer defined as boot/install clients in the SDR.

Example:

```
# delnimclient -l 7,8
```

This command deletes the NIM client definitions for nodes 7 and 8 from the NIM master.

2.2.9 allnimres

This wrapper is used to allocate the NIM resources to the clients depending on the value of the bootp_response attribute defined in the SDR. If the bootp_response value is set to install, migration, maintenance or diagnostic, the appropriate resources will be allocated. In case of a disk or customize value, all the resources will be deallocated.

Command syntax:

```
# allnimres -l <node_number_list>
```

`-l <node_number_list>` is the list of node numbers representing NIM clients to which to allocate resources. This is a required option.

Example:

```
# allnimres -l 5
```

This command allocates resources to node 5, depending on the value of the bootp_response attribute for node 5.

2.2.9.1 Resources Allocated for Various bootp_response Values

The resources that are allocated for performing various operations on a client are shown in the following screens. If you encounter problems in trying to perform any operation on the nodes from the NIM master, check if all these resources are allocated.

The command to check the resources that are allocated for performing a install operation on clients is as follows:

```
# lsnim -c resources sp4n09
boot                resources       boot
nim_script          resources       nim_script
psspscript          resources       script
noprompt            resources       bosinst_data
lppsource_aix432    resources       lpp_source
mksysb_1            resources       mksysb
spot_aix432         resources       spot
```

The resources that are allocated for performing a maintenance operation on clients are as follows:

```
# lsnim -c resources sp4n09
boot                resources       boot
nim_script          resources       nim_script
prompt              resources       bosinst_data
lppsource_aix432    resources       lpp_source
spot_aix432         resources       spot
```

The resources that are allocated for performing a diag operation on clients are as follows:

```
# lsnim -c resources sp4n09
boot                resources       boot
prompt              resources       bosinst_data
spot_aix432         resources       spot
```

The resources that are allocated for a migrate operation on clients are as follows:

```
# lsnim -c resources sp4n09
boot                resources       boot
nim_script          resources       nim_script
psspscript          resources       script
lppsource_aix432    resources       lpp_source
spot_aix432         resources       spot
migrate             resources       bosinst_data
```

If the bootp_response value is set to disk or customize no resources are allocated to the clients.

2.2.10 unallnimres

This wrapper is used to unallocate all the NIM resources for the clients. The command syntax is:

```
# unallnimres -l <node_number_list>
```

`-l <node_number_list>` is the list of node numbers representing NIM clients from which to deallocate resources. This is a required option.

Example:

```
# unallnimres -l 5
```

This command deallocates all resources for node 5.

Steps for Manually Configuring an Install Operation on a Node Using Wrappers

When you run `setup_server`, it looks for all the nodes and customizes them according to their bootp_response values each time you execute this command. This is really time-consuming and unnecessary. Instead of running `setup_server`, you can execute the following wrappers to set up the node for installation. The steps to set up node sp4n06 for installation are as follows:

```
# spbootins -s no -r install -l 6
```

This sets node 6 for the install operation and adds an entry in /etc/bootptab for node 6.

```
# create_krb_files
```

This creates and updates the /etc/tftpaccess.ctl and /tftpboot/sp4n06-new-srvtab files. It also sets the file permissions to read-only, and sets the owner of the file to user *nobody*.

```
# mkconfig
```

This creates the file /tftpboot/sp4n06.config_info.

```
# mkinstall
```

This command creates the file /tftpboot/sp4n06.install_info.

```
# export_clients
```

This command exports the directories for NFS mounts.

```
# allnimres -l 6
```

This command allocates all the resources that are required for the install operation for the node sp4n06.

Now from the perspectives, if you invoke a netboot on node 6, the node will reboot and start the installation of the node.

2.3 Web-Based System Manager for NIM (WSM)

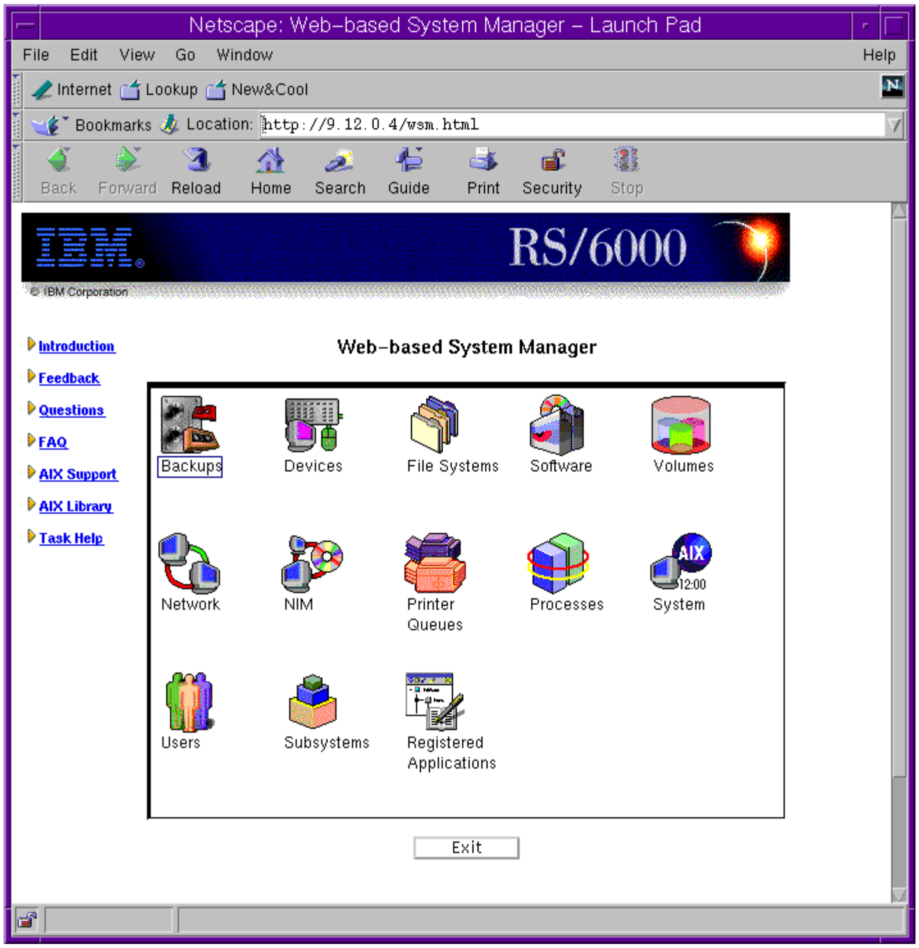
Web-Based System Manager (WSM) enables a system administrator to manage an AIX machine either locally from a graphics terminal or remotely by using a browser like Netscape or Internet Explorer. It has been implemented in the Java programming language.The Web browser in the client machine should be Java 1.1-enabled.

WSM includes the components that were available for doing system administration using SMIT. It can be launched from the Java-enabled browser and also from the application icon in the CDE application manager. To launch the WSM from a browser on a PC or some other client, you should have the Internet server daemon `httpd` running.

In this section we discuss how to do NIM administration using WSM. To open WSM from a Web browser, type the following in your browser's URL or location field:

```
http://<your server name>/wsm.html
```

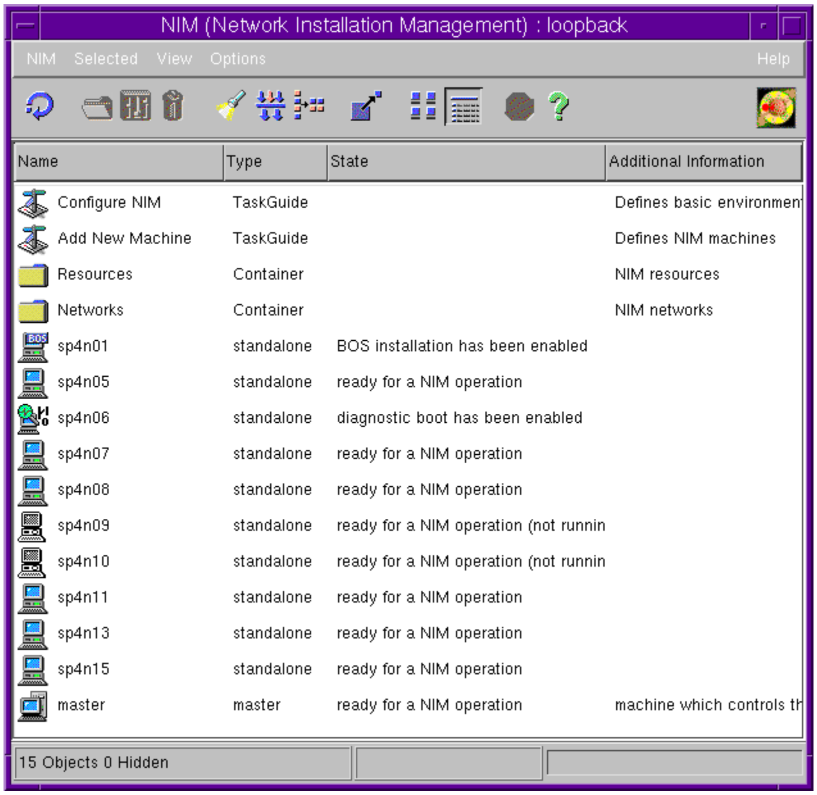
Figure 5 on page 42 shows the main screen when you open WSM from a browser.



Location: http://9.12.0.4/wsm.html

Figure 5. Web-Based System Manager Main Screen

When you double click on the NIM icon, it takes you to the NIM Installation Management menu, as shown in Figure 6 on page 43. This is the main screen for configuring and doing NIM administration.



Name	Type	State	Additional Information
Configure NIM	TaskGuide		Defines basic environmen
Add New Machine	TaskGuide		Defines NIM machines
Resources	Container		NIM resources
Networks	Container		NIM networks
sp4n01	standalone	BOS installation has been enabled	
sp4n05	standalone	ready for a NIM operation	
sp4n06	standalone	diagnostic boot has been enabled	
sp4n07	standalone	ready for a NIM operation	
sp4n08	standalone	ready for a NIM operation	
sp4n09	standalone	ready for a NIM operation (not runnin	
sp4n10	standalone	ready for a NIM operation (not runnin	
sp4n11	standalone	ready for a NIM operation	
sp4n13	standalone	ready for a NIM operation	
sp4n15	standalone	ready for a NIM operation	
master	master	ready for a NIM operation	machine which controls th

15 Objects 0 Hidden

Figure 6. Network Installation Management

There are three types of objects shown on this screen: the Task Guide, the Container, and the NIM machine objects (standalone and master). The screen also shows the state of each object, and for some objects it also gives additional information. For example, you can see that node sp4n01 is enabled for BOS installation, and that node sp4n06 is enabled for diagnostic boot.

Now let us describe these objects in detail.

TaskGuides:

These objects are dialogs designed to assist the user in performing tasks in an ordered series of steps. The first two lines in Figure 6 show two objects of type TaskGuide. They are:

Configure NIM

This TaskGuide helps you to configure the basic NIM environment. It initializes the host as a NIM master. You then have the option to create the NIM resources and the NIM clients in this NIM environment.

Add New Machine

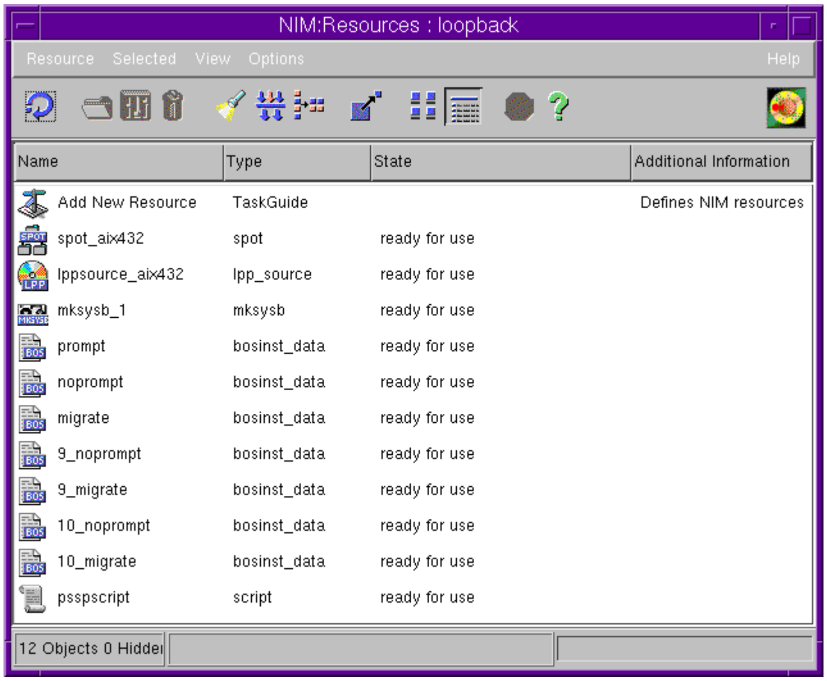
This is used for adding new standalone or diskless and dataless clients to the NIM environment. Here it is possible to specify multiple systems at the same time to define in the NIM database.

Container Objects:

The container objects include other container objects and simple objects representing elements of system resources to be configured and managed. In Figure 6 on page 43, you can see there are two types of container objects. They are:

Resources

This container object contains the resource objects that are configured in the NIM environment. Figure 7 on page 45 shows the NIM Resources screen.



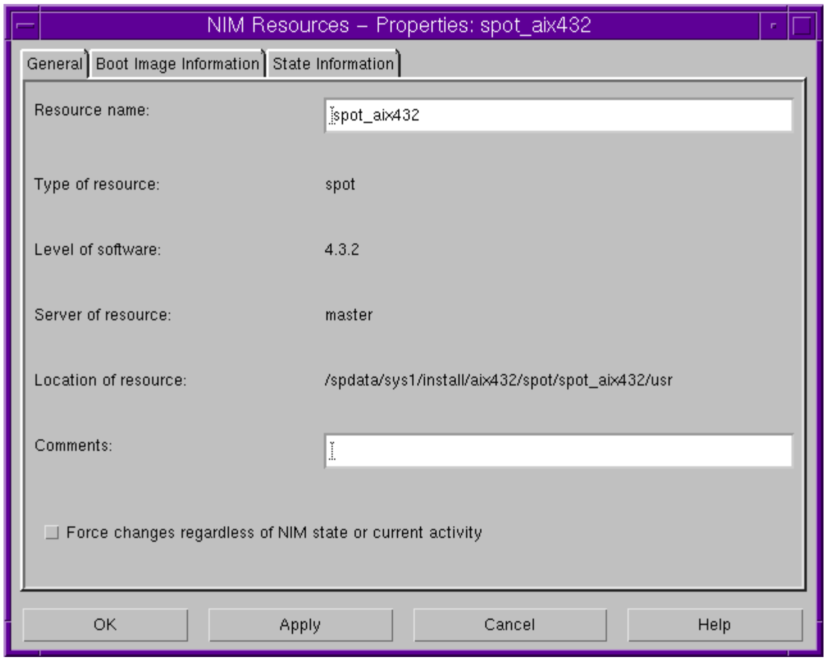
Name	Type	State	Additional Information
Add New Resource	TaskGuide		Defines NIM resources
spot_aix432	spot	ready for use	
lppsource_aix432	lpp_source	ready for use	
mksysb_1	mksysb	ready for use	
prompt	bosinst_data	ready for use	
noprompt	bosinst_data	ready for use	
migrate	bosinst_data	ready for use	
9_noprompt	bosinst_data	ready for use	
9_migrate	bosinst_data	ready for use	
10_noprompt	bosinst_data	ready for use	
10_migrate	bosinst_data	ready for use	
psspscript	script	ready for use	

12 Objects 0 Hidde

Figure 7. NIM Resources

The Add New Resource object is for creating new resources in the NIM environment. When you double-click on the other resource objects, it gives all the information and status of that resource.

For example, when you double-click on the spot_aix432 object, you will see the screen shown in Figure 8 on page 46. This screen shows the general information of this object, such as resource name, location of the resource, server of the resource and resource type. You also see that there are other menu options, like boot image information and the state information for this object.



Resource name: spot_aix432

Type of resource: spot

Level of software: 4.3.2

Server of resource: master

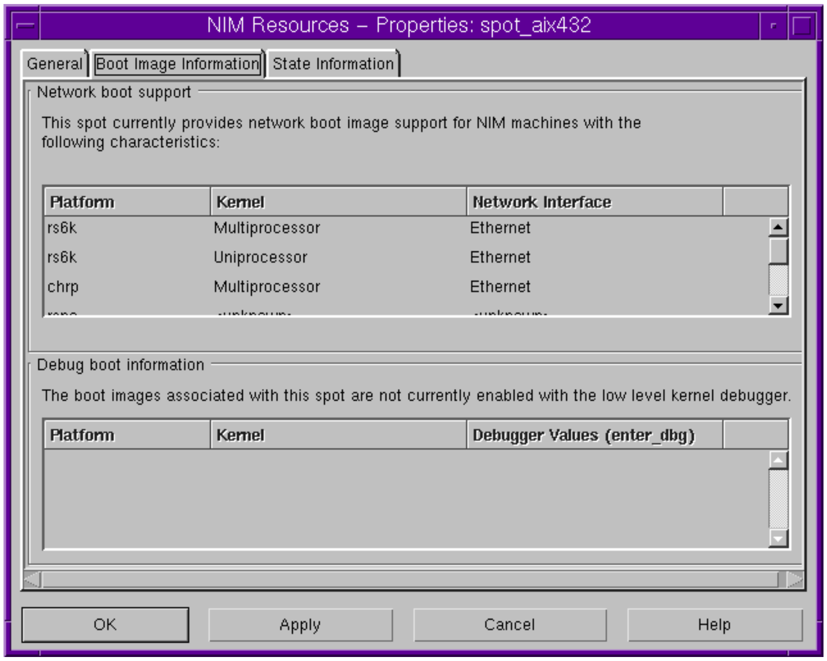
Location of resource: /spdata/sys1/install/aix432/spot/spot_aix432/usr

Comments:

□ Force changes regardless of NIM state or current activity

Figure 8. General Properties of the Resource spot_aix432

The boot image information of spot_aix432 is shown in Figure 9 on page 47. Here the display shows that the network boot image for a uniprocessor and multiprocessor system is configured.



This spot currently provides network boot image support for NIM machines with the following characteristics:

Platform	Kernel	Network Interface
rs6k	Multiprocessor	Ethernet
rs6k	Uniprocessor	Ethernet
chrp	Multiprocessor	Ethernet
rspc	<unknown>	<unknown>

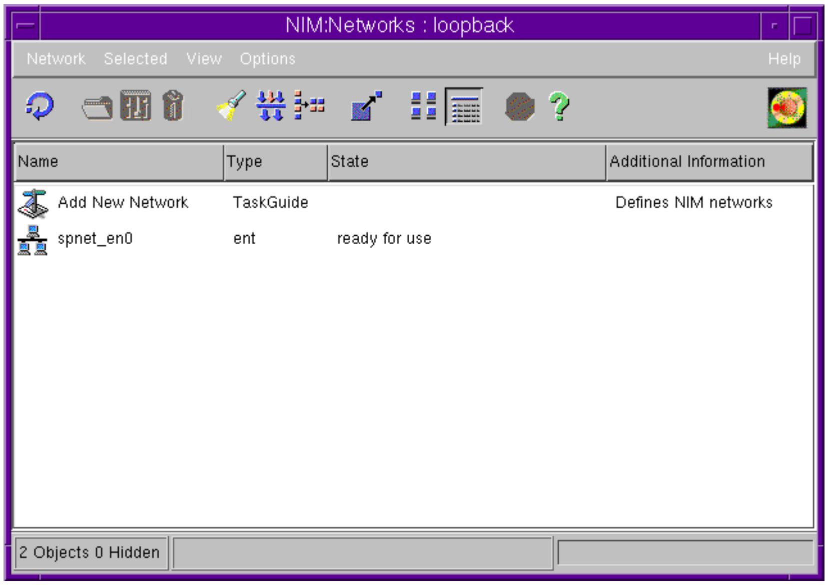
The boot images associated with this spot are not currently enabled with the low level kernel debugger.

Platform	Kernel	Debugger Values (enter_dbg)

Figure 9. Boot Image Information of the Resource spot_aix432

Networks

This network container object contains the Add New Network TaskGuide and the network object spnet_en0 that is configured in this NIM environment. Figure 10 on page 48 shows the NIM Networks screen.



Name	Type	State	Additional Information
Add New Network	TaskGuide		Defines NIM networks
spnet_en0	ent	ready for use	

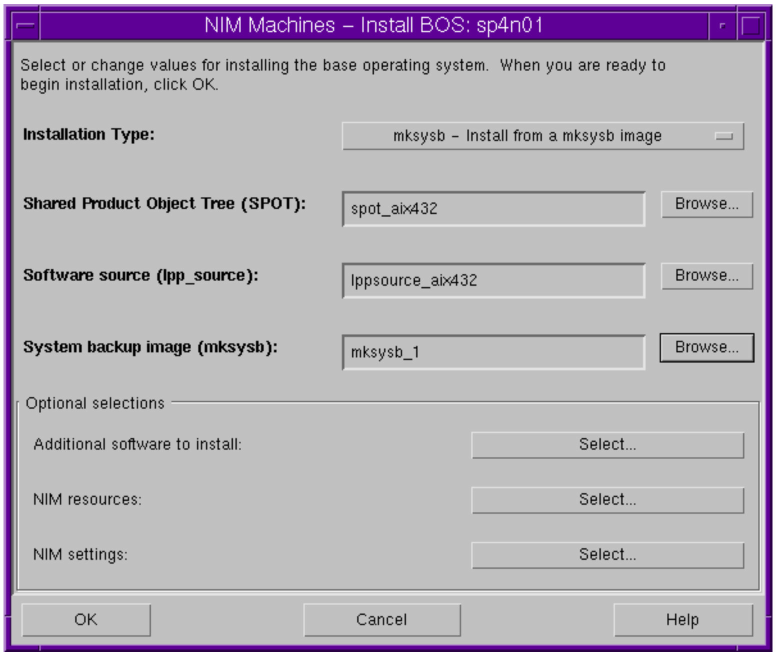
2 Objects 0 Hidden

Figure 10. NIM Networks

The Add New Network TaskGuide object is for creating new networks. Double-clicking the spnet_en0 object, you can see the general NIM routes and status information menus for this object.

Steps to perform a BOS Installation on sp4n10 Using WEBSM - NIM:

1. From the Network Installation Management screen, click the node **sp4n10** to see that it is highlighted.
2. Next click **Selected menu** at the top of the screen and you will see the various operations that can be performed on the clients.
3. Select the **Install Base Operating System** option, which opens another window, as shown in Figure 11 on page 49. This window is for allocating the resources and starting the bos_inst operation.



NIM Machines – Install BOS: sp4n01

Select or change values for installing the base operating system. When you are ready to begin installation, click OK.

Installation Type: mksysb – Install from a mksysb image

Shared Product Object Tree (SPOT): spot_aix432

Software source (lpp_source): lppsource_aix432

System backup image (mksysb): mksysb_1

Optional selections

Additional software to install:

NIM resources:

NIM settings:

Figure 11. BOS Installation for sp4n10

4. To perform the bos_inst operation using the mksysb_1 image, allocate the resources as shown in Figure 11 and click **OK** to start the installation.

2.4 The Installation Process

This section discusses the main steps involved in the installation process of the CWS and the nodes. Some steps are covered in more detail than others, simply because they are already detailed in other redbooks or installation guides. Where this is true, references to these other redbooks are made. This section can be used to diagnose and solve installation problems, or to simply verify that the installation process is running correctly.

The node installation process is complex, so we detail the process from start to finish. To help you find out what is happening during node installation, we also look at the debug options that are available.

At the start of this section, we look at the steps that are performed on the CWS before PSSP can be installed, and describe points that should be considered when preparing the CWS for an SP installation.

2.4.1 Setting Up the Control Workstation

After AIX has been successfully installed, you should read all the README documentation that arrived with your AIX and PSSP software and take the necessary actions before continuing with the installation.

2.4.2 Disk Space Considerations

We suggest that you install the /spdata filesystem in a separate volume group from rootvg, for the following reasons:

- Availability - if a disk fails, only one volume group is lost.
- You may want to install HACWS in the future. For this you would need to install spdata on an external disk subsystem.
- To save backup and restore time. The spdata filesystem is around 2 GB, and it can grow considerably with future releases of AIX and PSSP.
- You may already have free disk space limitations.

Following is an approximation of free space required in various filesystems and volume groups.

- The recommendation for rootvg is 2 GB, providing that /spdata is installed in a separate volume group.
- /var requires at least 20 MB of free space. Most of the PSSP logs are stored here. The actual disk space used by these logs depends entirely on the types of problems that occur on your system and their frequency. The /var filesystem should be monitored frequently to ensure that there is always sufficient free disk space for new logs.
- /tmp requires at least 16 MB of free space.
- We suggest that you create a separate file system for /tftpboot. Each lppsource level requires a minimum of 25 MB. You should also consider future AIX installations. We suggest 25 MB x (number of AIX versions+1). For example, if you are installing AIX 4.2, 4.2.1 and 4.3, then 25 MB x (3+1) = 100 MB.
- Downloading all of the AIX filesets to the lppsource directory requires approximately 1.5 GB of disk space. Downloading only the minimal filesets requires approximately 500 MB. Each mksysb image may vary between 100 MB and 700 MB. A simple example for AIX 4.3.2 is:

  lppsource + mksysb_images + pssp_lpp_images + SPOT = total disk space

  lppssource = 500 MB

mksysb_images = 300 MB

pssp_lpp_images = 350 MB

SPOT = 200 MB

500 MB + 300 MB + 350 MB + 200 MB = 1.35 GB

You should have at least 1.35 GB of free disk space for the /spdata file system. To be on the safe side, you should reserve 2 GB of disk space for /spdata. For multiple AIX and PSSP version coexistence, this figure will increase considerably, due to the disk space taken up by different versions of mksysb images, SPOT, AIX and PSSP filesets.

2.5 Directory Structure

Make sure that AIX and PSSP images are installed in the correct directory structure. See Figure 12 on page 52 for an example. Use the correct naming convention to store these images, otherwise the installation of the nodes will fail.

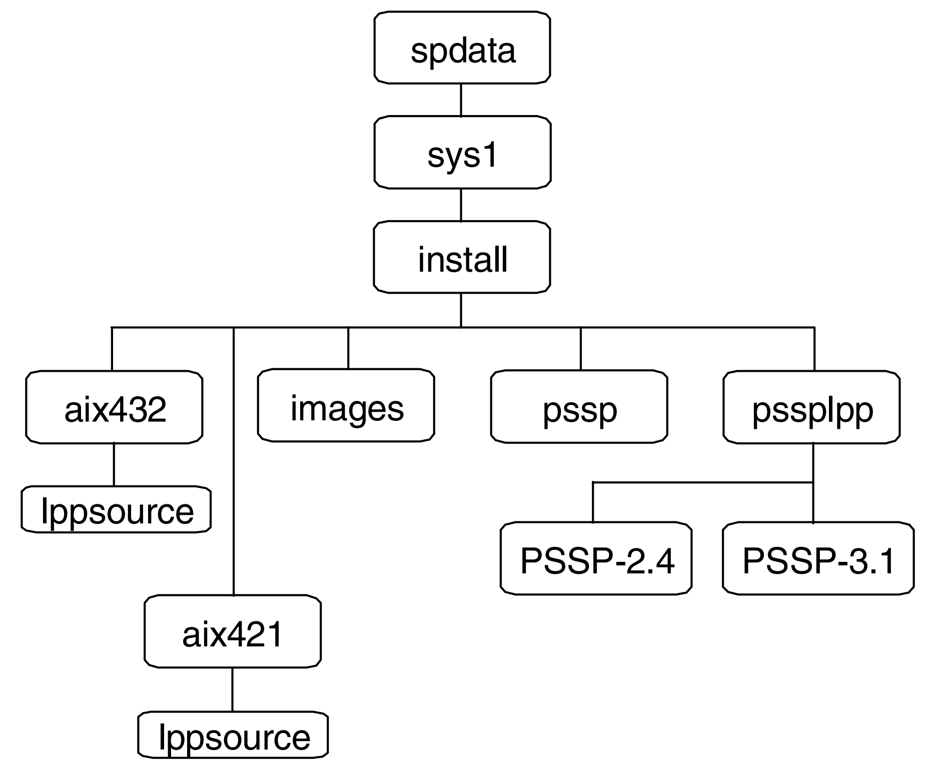


Figure 12. Multiple AIX and PSSP Levels

In an SP environment, for each AIX version an mksysb image, SPOT and lppsource must exist. For multiple AIX and PSSP installations, refer to Table 4 for supported versions.

Table 4. Supported AIX and PSSP Versions

		PSSP					AIX						
		2.1	2.2	2.3	2.4	3.1	4.1.4	4.1.5	4.2	4.2.1	4.3	4.3.1	4.3.2
PSSP	2.1	-	-	-	-	-	Y	Y	n	n	n	n	n
	2.2	Y	-	-	-	-	Y	Y	Y	Y	n	n	n
	2.3	Y	Y	-	-	-	n	n	n	Y	Y	Y	Y
	2.4	Y	Y	Y	-	-	n	n	n	Y	n	Y	Y
	3.1	Y	Y	Y	Y	-	n	n	n	n	n	n	Y

All AIX mksysb images must be installed in /spdata/sys1/install/images. The normal naming convention for mksysb images is bos.obj.ssp.<aix version>. In our case we installed AIX 4.3.2, which is bos.obj.ssp.432. However, you can choose your own name for the mksysb image. For multiple images, make sure the image name is version-specific.

All AIX filesets must be installed in lppsource using the directory structure shown in the following screen. The corresponding SPOT is also shown here. A SPOT resource contains basically all the files that are normally found in the /usr filesystem. When you install new filesets in lppsource, you should then update the corresponding SPOT; this is covered in 5.2, “Applying AIX PTFs in the CWS” on page 104.

```
/spdata/sys1/install/aix415/LPPsource
/spdata/sys1/install/aix415/spot
/spdata/sys1/install/aix421/LPPsource
/spdata/sys1/install/aix421/spot
/spdata/sys1/install/aix432/LPPsource
/spdata/sys1/install/aix432/spot
```

All PSSP filesets must be installed in /spdata/sys1/install/pssplpp/<code version>, as shown in the following screen.

```
/spdata/sys1/install/pssplpp/PSSP-2.2
/spdata/sys1/install/pssplpp/PSSP-2.3
/spdata/sys1/install/pssplpp/PSSP-2.4
/spdata/sys1/install/pssplpp/PSSP-3.1
```

Complete installation steps 1 to 17 described in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide,* GC23-3898.

You should now have completed the following:

1. AIX Installation
2. Required PTFs installed
3. Defined hostname, IP addresses, IP labels and netmasks
4. Verified network connections
5. Configured RS232 line between the CWS and frame(s)
6. Tuned CWS network adapters
7. Started the necessary daemons
8. Changed the CWS root Maximum Default Process value

9. Added network tunable values in /etc/rc.net

10. Created the /tftpboot filesystem

11. Created the /spdata filesystem (preferably in a separate volume group)

12. Created LPPsource and copied AIX filesets

13. Copied the PSSP image

14. Copied a basic AIX mksysb image

2.6 Additional Required AIX LPPs

Depending upon the level of AIX and PSSP you want to install, you have to order and install additional AIX LPPs. The following section gives an overview of which perfagent level is required and answers the question why a C-compiler is required.

The perfagent.server fileset is a prerequisite for PSSP. This fileset is part of the Performance Aide for AIX (PAIDE) feature of the Performance Toolbox for AIX (PTX), a separate product. The perfagent.tools fileset is part of AIX 4.3.2.

Table 5 gives an overview of the required level of perfagent.

Table 5. Perfagent Filesets

AIX Level	PSSP Level	Required Perfagent File Sets
4.1.5	2.2	perfagent.server 2.1.5.x
4.2.1	2.2	perfagent.server 2.2.1.x or greater, where x is greater than or equal to 2
4.2.1	2.3	perfagent.server 2.2.1.x or greater, where x is greater than or equal to 2
4.2.1	2.4	perfagent.server 2.2.1.x or greater, where x is greater than or equal to 2
4.3.1	2.3	perfagent.server 2.2.31.x
4.3.1	2.4	perfagent.server 2.2.31.x
4.3.2	2.3	perfagent.tools and perfagent.server 2.2.32.x
4.3.2	2.4	perfagent.tools and perfagent.server 2.2.32.x
4.3.2	3.1	perfagent.tools 2.2.32.x

Another prerequisite for PSSP is IBM C for AIX. This LPP is necessary for service of the PSSP. Without the compiler’s preprocessor, dumb diagnostic tools such as crash will not function fully. At least a one-user license is required.

For an installation of AIX 4.3.2 and PSSP 3.1 the following filesets need to be copied to /spdata/sys1/install/aix432/lppsource:

- xlC.rte 3.6.4.0
- perfagent.tools 2.2.32.x

2.7 Minimum Required PSSP Filesets

These are the minimum required PSSP filesets:

- rsct.* (all rsct filesets)
- ssp.basic
- ssp.authent (if CWS is a Kerberos authentication server)
- ssp.clients
- ssp.css (if SP Switch is installed)
- ssp.top (if SP Switch is installed)
- ssp.ha_topsvcs.compat
- ssp.perlpkg
- ssp.sysctl
- ssp.sysman

Note

If you are adding dependent nodes, you must also install the ssp.spmgr fileset.

2.8 Install PSSP on the Control Workstation

Install PSSP on the CWS, either from media or the PSSP images that were earlier copied onto the CWS.

Install the PSSP software from SMIT panels, or use the `installp` command.

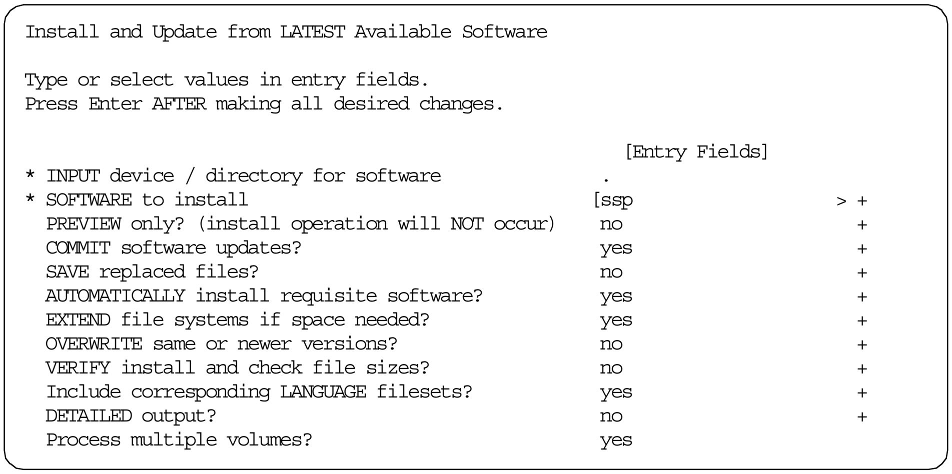
To install PSSP from the command line, enter the following command:

```
# installp -a -d /spdata/sys1/install/pssplpp/<PSSP-VER> -X ssp.*
```

Alternatively, use the SMIT panels to perform the PSSP installation, as follows:

```
# cd /spdata/sys1/install/pssplpp/PSSP-3.1

# smitty install_latest
```

Select the current directory as your input device.



```
Install and Update from LATEST Available Software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* INPUT device / directory for software                 .
* SOFTWARE to install                                  [ssp                  > +
  PREVIEW only? (install operation will NOT occur)      no                     +
  COMMIT software updates?                              yes                    +
  SAVE replaced files?                                  no                     +
  AUTOMATICALLY install requisite software?             yes                    +
  EXTEND file systems if space needed?                  yes                    +
  OVERWRITE same or newer versions?                     no                     +
  VERIFY install and check file sizes?                  no                     +
  Include corresponding LANGUAGE filesets?              yes                    +
  DETAILED output?                                      no                     +
  Process multiple volumes?                             yes
```

Once PSSP has been installed on the CWS, you can perform the rest of the steps required to configure our CWS as an authentication server, NIM master and boot/install server. These steps need to be performed before you can install the SP nodes.

2.9 setup_authent

To proceed with the rest of the installation, you need to create a primary authentication server, which is normally the CWS. The minimum configuration required at this stage is to initialize the Kerberos database and define and register a kerberos administrative user. The administrative UNIX user is root. A Kerberos principal must be created (it should be named root.admin) before you can perform subsequent steps in the installation. Enter the following command to configure the Kerberos database and to set up the root.admin principal:

```
# setup_authent
```

The Creating the Kerberos Database menu will appear. Answer the prompts to initialize the primary authentication server and get a Kerberos ticket for the Kerberos administration principal root.admin.

If you intend to install a secondary authentication server, follow the steps in 8.12, “Setting Up a Secondary Authentication Server” on page 197, once the primary server has been initialized. Note that you need to ensure clock synchronization between primary and secondary authentication servers.

2.9.1 setup_authent Files and Daemons

The setup_authent script creates the following files and daemons, which are discussed in detail in 8.4, “Kerberos Daemons” on page 163.

Files

/.k	master key cache file
/.klogin	lists the names of principals authorized to invoke remote commands
/etc/krb.conf	contains the local realm and a list of authentication servers for the realm
/etc/krb.realms	maps host names to an authentication realm
/etc/krb-srvtab	this server key file contains names and private keys
/tmp/tkt0	this ticket cache file contains the ticket granting ticket
/var/kerberos/database/principle.dir	part of the Kerberos database
/var/kerberos/database/principle.pag	part of the Kerberos database

Daemons

The setup_authent script starts the kerberos and kadmind daemons. These daemons are added to the /etc/inittab file and are started from here thereafter. The following screen shows an entry for each of the kerberos daemons added to the /etc/inittab file.

```
# cat /etc/inittab

kerb:2:once:/usr/bin/startsrc -s kerberos
kadm:2:once:/usr/bin/startsrc -s kadmind
```

The kerberos daemon is responsible for providing ticket granting tickets to clients. The kadmind daemon is responsible for the administration of the

Kerberos principals, typically for adding principals and changing principal passwords.

If you were installing at least one secondary authentication server, there would be an entry in the inittab file for the kpropd daemon on the secondary authentication server.

The kpropd daemon only exists on the secondary authentication server and is responsible for updating the secondary authentication server's database from the primary authentication server.

2.10 The install_cw Command

Enter the `klist` command to ensure that you have a valid ticket granting ticket (TGT) for the root.admin principal. The following screen shows the command output.

```
# /usr/kerberos/bin/klist
Ticket file:    /tmp/tkt0
Principal:      root.admin@SP4EN0

  Issued           Expires          Principal
Apr 28 18:08:23  May 28 18:08:23  krbtgt.SP4EN0@SP4EN0
```

If your ticket is expired, you have to run `kinit root.admin` from the command line and enter your password. Once you have a ticket granting ticket, enter the following command:

```
# install_cw
```

This script performs the following steps:

1. It creates the /spdata/sys1/spmon/hmacl file.

   The hmacl file contains the permissions specifications for Kerberos principals. The following screen shows the default entries for root.admin and hardmon. Both principals have vsm permissions. What this actually means is that both principals can view, save, and modify.

   ```
   # cat /spdata/sy1/spmon/hmacl

   1 root.admin vsm
   1 hardmon.sp4en0 vsm
   ```

2. It creates the /etc/SDR_dest_info file. This file contains the IP address and name of the SDR server, which is normally the CWS. The following screen shows the output of our SDR_dest_info file.

```
# cat /etc/SDR_dest_info

default:192.168.4.140
primary:192.168.4.140
nameofdefault:sp4en0
nameofprimary:sp4en0
```

3. It creates the sdrd and hardmon subsystems.
4. It creates an entry for the sdrd in /etc/inittab.
5. It creates an entry for the hardmon daemon in /etc/inittab.
6. It creates the following port map entries in the /etc/services:

```
hardmon 8435/tcp

sdr 5712/tcp

heartbeat 4893/udp
```

2.11 The setup_server Command

The `setup_server` command calls a number of scripts and wrappers to perform the following task. These wrappers are discussed in detail in 2.2, “NIM Configuration Using PSSP” on page 31.

The main tasks performed by the `setup_server` command are setting up PSSP services such as NTP, AMD and File Collections. This command ensures that the required Kerberos files exist and then creates a list of known rcmd principals. It creates the CWS as a NIM master and gets information from the SDR to create the NIM environment including network objects, NIM clients, SPOT and tftp boot images for the different platforms, and stores them in the /tftpboot directory. These files are discussed in detail in 2.14.1, “/tftpboot Files” on page 63.

After creating the NIM environment, the NIM resources are allocated to NIM clients inside the SP. When `setup_server` is run for the first time on a CWS, all of these steps need to be performed. This can take up to one hour or more depending on the CWS configuration and hardware environment. Creating the SPOT takes up the majority of this time.

NIM logs the actions performed during SPOT creation in the /tmp directory. There you should look for the file name spot.out.<pid>; where pid is the process ID of the SPOT creation process. This process is probably not running any more, so in case you find several spot.out.xxx files, take the newest one. View the contents of this file and see if you find any hints why the build of the SPOT was unsuccessful.

2.12 Verification Tests

Upon the completion of `setup_server` you should reboot the CWS and then perform configuration verification and verification tests to ensure that the CWS is set up correctly. Verify that all the necessary daemons are running. Run the following tests:

```
# SDR_test
```

The SDR_test creates an SDR class and adds attributes and values, then deletes the SDR class to ensure the SDR is working correctly.

```
# spmon_itest
```

The spmon_itest verifies the following:

- The SP_ports class in the SDR contains hardmon information.
- The hardmon daemon is running.
- The sdrd daemon is running.
- It validates the hmacls file.

After these tests are complete, log files are created in /var/adm/SPlogs: SDR_test.log and spmon_itest.log.

2.13 Debugging NIM

If the SP node was booted from the network boot image, but failures are still occurring during a BOS installation, it may be necessary to collect debug information from the BOS install program. This is achieved by building *debug boot images* using the SPOT. After doing this, commands and output produced by the BOS install program will automatically be displayed on the tty session connecting to the node. Typical symptoms of BOS installation problems are system hangs.

Viewing the debug output can be extremely useful, because you will be able to see the commands that failed. The problem may be a misconfiguration of the boot network adapter, incorrect NIM configuration information, or errors in resource definitions. Examining the debug output, you can reduce the scope of investigation, isolate the cause of the problem, and fix it.

Perform the following steps to produce debug output from a network boot image:

Step 1: Create the debug boot images
To create debug versions of the network boot images use the `nim` command on the NIM master:

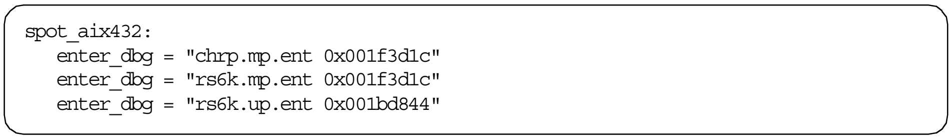
```
# nim -Fo check -a debug=yes SPOTName
```

where SPOTName is the name of your SPOT (for example, spot_aix432).

Step 2: Get the address used for the debug session
On the master, use the following command:

```
# lsnim -a enter_dbg SPOTName
```

where SPOTName is the name of your SPOT (for example, spot_aix432). The displayed output will be similar to the one shown in the following sample:



```
spot_aix432:
   enter_dbg = "chrp.mp.ent 0x001f3d1c"
   enter_dbg = "rs6k.mp.ent 0x001f3d1c"
   enter_dbg = "rs6k.up.ent 0x001bd844"
```

Figure 13. SPOT Platform-Dependent Debug Addresses

Write down the enter_dbg address for the type of node you are going to boot. For example, if your client is an rs6k multiprocessor machine, you would write down the address 1f3d1c.

You can query the platform and processor type information (for example: node #1) by using the following command:

```
# SDRGetObjects -x Node node_number==1 platform processor_type
```

The CWS responds with:

```
rs6k         MP
```

Step 3: Open a terminal session to the node
You have to open a read/write terminal session to the node, because the system prompts you during the installation. You also want to keep track of the debug information using a file named /tmp/BOS.trace.n01. To access node #1 enter:

```
# s1term -w 1 1 | tee /tmp/BOS.trace.n01
```

Press Enter twice.

Step 4: Netboot the node

We assume that the node is set to customize and the `setup_server` script was executed.

The boot process has to be initiated using the `spmon` command. To boot node 1, enter:

```
# spmon -power on node1
```

During the boot process a menu will appear and the system will prompt you for the selection of the network boot device. Make sure to select the SP Ethernet adapter. This is “normally” automated by the `nodecond` command during node installation. Now it has to be done “manually”.

Step 5: Enter debug information

After the client gets the boot image from the NIM master, the debug screen appears on the terminal. You see some cryptic information and the ">" prompt will be shown. Enter:

```
st Enter_dbg_Value 2
```

where `Enter_dbg_Value` is the number you wrote down in step 2 as your machine type's address (in our example, 1f3d1c). Specifying the `2` in the end of your command makes sure output is printed to your tty.

After getting the prompt again, type `g` for go and press Enter to start the boot process.

Use Ctrl-s to temporarily stop the process as you watch the output on the tty. Use Ctrl-q to resume the process.

The system will begin installation and provide you with the commands and outputs during the installation process. If the system hangs, you can spot the command causing this and what the problem is.

Step 6: Build a normal SPOT

After a successful debug session rebuild your SPOT in non-debug mode. Use the following command on the NIM master:

```
# nim - Fo check SPOTName
```

where SPOTName is the name of your SPOT (for example, spot_aix432).

If the boot image is left in debug mode, then every time a client is booted from these boot images, the machine will stop and wait for a command at the debugger ">" prompt. If you attempt to use these debug-enabled boot images and there is not a tty attached to the client, the machine will appear to be hanging for no reason.

2.14 Set bootp_response to Install

In this example we are going to install node sp4n09 with AIX 4.3.2 and PSSP 3.1. We use hdisk0 for rootvg. From either SMIT or the command line, set the node to install. From the command line enter the following command:

```
# spbootins -c rootvg -r install -l 9 -s yes
```

Enter the following command to verify that the node boot response is set to install. Check that the version levels for AIX and PSSP software are correct. Check the hdisk number for the boot disk.

```
# splstdata -b -l 9
```

```
 [sp4en0:/tmp]# splstdata -b -l 9
                  List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr      response            install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
--------------------------------------------------------------------------------
    9 sp4n09.msc.itso.   0004AC4947E9    0         disk                   hdisk0
                 default Fri_Apr__9_18:26:04             default          aix432
                PSSP-3.1              rootvg
```

2.14.1 /tftpboot Files

When a node is set to install and `setup_server` is run on the CWS, the setup_server script updates the /etc/tftpaccess.ctl file on the server. In the /tftpboot directory the following files are created for sp4n09:

sp4n09-new-srvtab

sp4n09.config_info

sp4n09.install_info

Each file name starts with the reliable hostname. The srvtab file is the Kerberos service key file; for more information, see 8.6, “The krb-srvtab File” on page 166.

The config_info file is described in 2.2.4.1, “The config_info File” on page 33 and the install_info is described in 2.2.5.1, “The install_info File” on page 34. They are created by the mkinstall and mkconfig wrappers. For more information on these wrappers, see 2.2.4, “mkconfig” on page 33.

An entry for node sp4n09 is added to the bootpd configuration file /etc/bootptab on the CWS. Each bootpd configuration parameter in this single line (in our screen output this line was broken down into two lines) is two characters long and is separated from the other parameters. The following screen shows an example entry for one node:

```
cat /etc/bootptab

sp4n09:bf=/tftpboot/sp4n09:ip=192.168.4.9:ht=ethernet:ha=10005AFA158A:sa=
192.168.4.140:sm=255.255.255.0:
```

Here is an explanation of each parameter:

- The first field specifies the node name, which in this example is sp4n09, followed by a character tag bf.
- bf specifies the bootfile /tftpboot/sp4n09.
- ip specifies the ip address of the node.
- ht specifies the network type, which is always Ethernet for SP nodes.
- ha specifies the hardware address of the Ethernet adapter.
- sa specifies the ip address of the TFTP server.
- sm specifies the host subnet mask.

All the NIM resources required for an install bootp response are created and allocated to the node (for example the bos image, SPOT and lppsource to use). NIM resources for PSSP script and customization scripts are also allocated. Enter the following command to verify the NIM resources allocated to the node:

```
# lsnim -l sp4n09
```

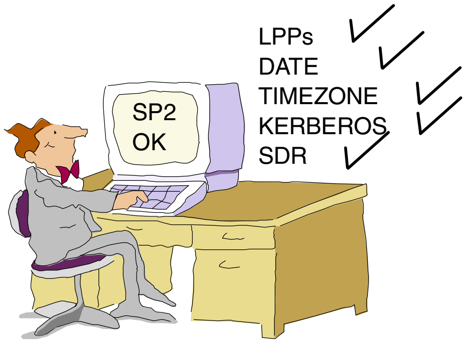
You should get a response similar to the following:

```
class          = machines
type           = standalone
platform       = rs6k
netboot_kernel = up
if1            = spnet_en0 sp4n09 10005AFA158A ent
cable_type1    = bnc
Cstate         = BOS installation has been enabled
prev_state     = ready for a NIM operation
Mstate         = currently running
boot           = boot
bosinst_data   = noprompt
lpp_source     = lppsource_aix432
mksysb         = mksysb_1
NIM_script     = NIM_script
script         = psspscript
spot           = spot_aix432
cpuid          = 000201335700
control        = master
```

Chapter 3. Verifying Your SP

An SP system is a complex interrelation among the many components that make this machine powerful and interesting. To ensure that your system is working properly, and to avoid future problems, you should verify from the very beginning that all components of your SP system are working as expected and will be available when needed.

You should not underestimate the importance of performing verification procedures at the time of installation. Just as you would not wait until you were going 70 mph before testing the brakes on a new car, or wait until dark before testing its headlights, neither should you wait until your system is in production (and perhaps running into problems) before performing verification.



This chapter provides guidelines that you can use to verify that your SP system is well installed and customized, and that all subsystems integrated in the SP are running as needed according to your specific environment. Some scripts are also provided through the SMIT panels to verify your system. You can type `smitty sp_verify` to go to the panel of verifications and execute the scripts.

It is also helpful if you perform these verifications whenever you finish executing any procedure that changes your environment (for example, initial installation, migration, installation of a new node, installation of PTFs, partitioning, addition of a new switch and so on).

Following the verification procedures in the order in which they are presented in this chapter is a good way to verify your whole system. However, you can also use them when only verifying some of your subsystems.

While the purpose of this chapter is to verify your SP system and isolate problems, refer to *RS/6000 SP: Problem Determination Guide,* SG24-4778 for problem resolution procedures.

3.1 LPP Level

The level of the LPPs should be the latest available at the time of your installation. Try to obtain that level at the start because some PTFs require you to do specific procedures to be completely installed, such as taking down your nodes, or some of the subsystems, or the Control Workstation (CWS). Activities like rebooting a node may interrupt the production system and are therefore difficult to do on a working system. For example a PTF for the switch could, in some cases, cause the switch to stop.

Never go to production having just the software included in the original media, because many problems are found after the software is installed and used massively by the customers.

There are also some specific PTFs required for AIX that should be installed in the CWS and the nodes so they will function properly. These PTFs are mentioned in the README FIRST files, so be sure to read that documentation to certify that all the PTFs are installed in your system. You can verify the level of your SP software using the following command:

```
# lslpp -L | grep -E "ssp|rsct"

rsct.basic.hacmp          3.1.0.2    A    RS/6000 Cluster Technology basic
rsct.basic.rte            3.1.0.4    A    RS/6000 Cluster Technology basic
rsct.basic.sp             1.1.0.3    A    RS/6000 Cluster Technology basic
rsct.clients.hacmp        1.1.0.0    C    (ECIP) RS/6000 Cluster
rsct.clients.rte          1.1.0.2    A    RS/6000 Cluster Technology
rsct.clients.sp           1.1.0.0    C    (ECIP) RS/6000 Cluster
ssp.authent               3.1.0.1    A    SP Authentication Server
ssp.basic                 3.1.0.4    C    SP System Support Package
ssp.clients               3.1.0.4    C    SP Authenticated Client Commands
ssp.css                   3.1.0.4    C    SP Communication Subsystem
ssp.ha_topsvcs.compat     3.1.0.0    C    Compatability for ssp.ha and
                                                 ssp.topsvcs clients
ssp.docs                  3.1.0.0    C    SP man pages and PDF files and
ssp.gui                   3.1.0.4    C    SP System Monitor Graphical User
ssp.jm                    3.1.0.1    C    SP Job Manager Package
ssp.perlpkg               3.1.0.0    C    SP PERL Distribution Package
ssp.pman                  3.1.0.1    A    SP Problem Management
ssp.ptpegui               3.1.0.1    C    SP Performance Monitor Graphical
ssp.public                3.1.0.0    C    Public Code Compressed Tarfiles
ssp.resctr.rte            3.1.0.0    C    SP Resource Center
ssp.spmgr                 3.1.0.2    A    SP Extension Node SNMP Manager
ssp.st                    3.1.0.1    A    Job Switch Resource Table
ssp.sysctl                3.1.0.1    A    SP Sysctl Package
ssp.sysman                3.1.0.2    A    Optional System Management
ssp.tecad                 3.1.0.0    C    SP HA TEC Event Adapter Package
ssp.top                   3.1.0.1    A    SP Communication Subsystem
ssp.top.gui               3.1.0.1    C    SP System Partitioning Aid
ssp.ucode                 3.1.0.1    A    SP Supervisor Microcode Package
ssp.vsdgui                3.1.0.1    C    VSD Graphical User Interface
```

There are some LPPs that are absolutely necessary for your CWS; the following list specifies the minimum PSSP LPPs that you should have installed:

rsct.basic.hacmp

rsct.basic.rte

rsct.basic.sp

rsct.clients.hacmp

rsct.clients.rte

rsct.clients.sp

ssp.authent (if CWS is Kerberos authentication server)

ssp.basic

ssp.clients

ssp.css (if switch is installed)

ssp.ha_topsvcs.compat

ssp.perlpkg

ssp.sysct

ssp.sysman

ssp.top (if switch is installed)

ssp.ucode

If you are using Virtual Shared Disks (VSDs), you should see:

vsd.cmi

vsd.hsd

vsd.rvsd.hc

vsd.rvsd.rvsdd

vsd.rvsd.scripts

vsd.sysctl

vsd.vsdd

This list is valid for PSSP 3.1. It may vary for other PSSP versions.

Other products that are included with the PSSP media, but are not absolutely necessary unless you have that specific requirement, are:

ssp.docs

ssp.gui

sss.hacws

ssp.pman

ssp.ptpegui

ssp.public

ssp.resctr.rte

ssp.st

ssp.tecad

ssp.top.gui

ssp.vsdgui

After you verify that all the required LPPs are installed, check for the required PTFs. Use the `instfix` command to determine weather a specific APAR is installed. For example, to display if APAR IX71835 is installed in your system, type:

```
# instfix -ik IX71835
```

You will have one of two responses:

```
All filesets for IX71835 were found.
```

Or:

```
There was no data for IX71832 in the fix database.
```

If you want a complete display of this fileset, type:

```
#insfix -ivk IX71835
```

The response will be:

```
   IX71835 Abstract: nfs v3 sequential write performance
   Fileset bos.mp is not applied on the system.
   Fileset bos.net.nfs.cachefs:4.3.0.1 is applied on the system.
   Fileset bos.net.tcp.client:4.3.0.1 is applied on the system.
   Fileset bos.up:4.3.0.1 is applied on the system.
   Fileset bos.adt.include:4.3.0.1 is applied on the system.
   All filesets for IX71835 were found.
```

After you have received this response, you should verify that the software is consistent throughout your system; that is, that your nodes and CWS have the same levels of software.

There are different methods of performing this verification:

- Locally: If you have only a few nodes, you can do this verification by logging in and executing the same command that you executed before and comparing the results with the ones from the CWS.
- Remotely: You can do this verification by executing the command `lppdiff` with flags such as the following:

-a Specifies that the System Data Repository (SDR) initial_hostname field for all nodes in the current system partition be added to the working collective. If -G is also specified, all nodes in the SP system are included.

-c Displays information as a list separated by colons.

-G Expands the scope of the -a flag to include all nodes in the SP system. The -G flag is meaningful only if used in conjunction with the -a flag.

-n Displays the count of the number of nodes with a fileset.

-v Verifies hosts before adding to the working collective. If this flag is set, each host to be added to the working collective is checked before being added.

-w Specifies a list of host names, separated by commas, to include in the working collective. Both this flag and the -a flag can be included on the same command line. Duplicate host names are included only once in the working collective.

For example, to query LPP information for ssp.basic on all nodes in the system, enter:

```
# lppdiff -Ga ssp.basic
```

You should receive output similar to the following:

```
-------------------------------------------------------------------------------
      Name         Path                Level        PTF     State     Type  Num
-------------------------------------------------------------------------------
LPP:  ssp.basic    /etc/objrepos       3.1.0.0              COMMITTED I      10
From: sp4n01 sp4n05 sp4n06 sp4n07 sp4n08 sp4n09 sp4n10 sp4n11 sp4n13 sp4n15
-------------------------------------------------------------------------------
LPP:  ssp.basic    /etc/objrepos       3.1.0.4              APPLIED   F      10
From: sp4n01 sp4n05 sp4n06 sp4n07 sp4n08 sp4n09 sp4n10 sp4n11 sp4n13 sp4n15
-------------------------------------------------------------------------------
LPP:  ssp.basic    /usr/lib/objrepos   3.1.0.0              COMMITTED I      10
From: sp4n01 sp4n05 sp4n06 sp4n07 sp4n08 sp4n09 sp4n10 sp4n11 sp4n13 sp4n15
-------------------------------------------------------------------------------
LPP:  ssp.basic    /usr/lib/objrepos   3.1.0.4              APPLIED   F      10
From: sp4n01 sp4n05 sp4n06 sp4n07 sp4n08 sp4n09 sp4n10 sp4n11 sp4n13 sp4n15
```

If you find any inconsistency with the software, stall the required LPPs or PTFs.

3.2 Dates and Timezones

In a “normal” machine, having the correct date is important, but in the SP it is vital. However, it is easy to check the correct date and thereby avoid problems in your environment. Simply make sure that the CWS has the correct date and also check the specified timezone, as shown:

```
# date
Fri Apr 16 17:19:27 EDT 1999
```

Do the same with the nodes. You can use the dsh command for this check:

```
# dsh -a date
```

If you see a difference of more than five minutes or the timezone is incorrect, you will have to change it because there are daemons and subsystems that use time stamps to accomplish their work. For example, kerberos uses a time stamp when decoding tickets for authentication, and cannot handle differences of more than five minutes in the clocks of the authenticator and the clients.

If you find such differences in your system, refer to *IBM Parallel System Support Programs for AIX: Administration Guide,* GC23-3897 to understand the procedure to change hostnames and IP addresses.

3.3 Checking Kerberos

Kerberos is the authentication subsystem used in the SP. It is based on Version 4 of the Kerberos authentication service developed at MIT. Kerberos validates the identity of either a user of a service, or the service itself.

For a fast check of kerberos, type the following command:

```
# dsh -a date
```

You should receive responses from all nodes that are powered on, giving you their system date, as in the following example:

```
# dsh -a date
sp4n01: Fri Apr 16 10:55:02 EDT 1999
sp4n05: Fri Apr 16 10:55:02 EDT 1999
sp4n06: Fri Apr 16 10:55:02 EDT 1999
```

If you have an improper configuration, you will receive messages such as:

```
sp4n10.msc.itso.ibm.com: Fri Apr 16 10:36:44 EDT 1999
sp4n10.msc.itso.ibm.com: rshd: Kerberos Authentication Failed: Access
denied because of improper credentials.
sp4n10.msc.itso.ibm.com: spk4rsh: 0041-004 Kerberos rcmd failed: rcmd
protocol failure.
sp4n11.msc.itso.ibm.com: Fri Apr 16 10:36:44 EDT 1999
sp4n11.msc.itso.ibm.com: rshd: Kerberos Authentication Failed: Access
denied because of improper credentials.
sp4n11.msc.itso.ibm.com: spk4rsh: 0041-004 Kerberos rcmd failed: rcmd
protocol failure.
```

If you receive an error in that command, verify that your tickets are available and have not expired by typing `klist`.

```
Ticket file:    /tmp/tkt0
Principal:      root.admin@SP4EN0

  Issued           Expires          Principal
Apr 15 11:31:24  May 14 11:31:24  krbtgt.SP4EN0@SP4EN0
Apr 15 11:31:24  May 14 11:31:24  rcmd.sp4n06@SP4EN0
Apr 15 11:31:24  May 14 11:31:24  rcmd.sp4n01@SP4EN0
Apr 15 11:31:24  May 14 11:31:24  rcmd.sp4n08@SP4EN0
Apr 15 11:31:24  May 14 11:31:24  rcmd.sp4n07@SP4EN0
Apr 15 11:31:24  May 14 11:31:24  rcmd.sp4n05@SP4EN0
```

Check the column labeled Expires*.* If you need to update your tickets, execute the command `kinit` followed by the principal, as in this example:

```
# kinit root.admin
```

Within the Kerberos subsystem, you should verify that the following daemons are running and that the following files exist, with the correct data and permissions.

3.3.1 Kerberos and kadmind Daemons

These daemons should be running in the primary authentication server (usually the CWS). The kerberos daemon provides ticket-granting tickets to clients so that they can establish communication with the server.The kadmind daemon manages the administrative tools of the kerberos database.

To verify that these daemons are running in the CWS, execute:

```
# ps -fea | grep kerb
root 12688  4906   0   Apr 02      -  0:00 /usr/lpp/ssp/kerberos/etc/kadmind -n
root 13724  4906   0   Apr 02      -  0:00 /usr/lpp/ssp/kerberos/etc/kerberos
```

You can also check that these daemons are in the /etc/inittab file so they will be started automatically in each reboot.

3.3.2 Kerberos Files

To do a more in-depth verification, you can check the eight files related to kerberos that must exist in your CWS and nodes (except for the*.k* file, which will be only in your CWS).

/.k

This file contains the master key of the kerberos database. The kadmind and the utility commands read the key from this file instead of prompting for the master password.

The permissions are: -rw------ , owner root and group system.

$HOME/.klogin

This file specifies a list of the remote principals that are authorized to invoke commands on the local user account. For example, the .klogin file of the user root file contains the principals who are authorized to invoke processes as the root user with the kerberos remote commands (rsh and rcp).The .klogin file of root is distributed to the nodes during installation or customization.

The permissions are: -rw-r--r-- owner root and group system.

/tmp/tkt<uid>

This file contains the tickets owned by a client of the authentication database. This file is continuously created and destroyed using the `kinit` and `kdestroy` commands.

To verify that you have correct tickets, you can reinitialize the file by executing the `kinit` command. You should be prompted for the password of the principal and if it is correct, you should return to the AIX prompt, as follows:

```
# kinit root.admin
Kerberos Initialization for "root.admin"
Password:
```

The permissions are: -rw------ , owner <uid> and group <group of uid>.

/etc/krb-srvtab

This file contains the names and private keys of the local instances for all the services protected by Kerberos. It is important to verify that the key versions of the nodes match the ones specified in the CWS for the same services. To check this, execute:

```
# klist -srvtab
```

In the CWS, you will have a response similar to:

```
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
------------------------------------------------------
rcmd            sp4en0          SP4EN0          1
hardmon         sp4en0          SP4EN0          1
```

In the nodes, you will have a response similar to:

```
Server key file:   /etc/krb-srvtab
Service         Instance        Realm      Key Version
------------------------------------------------------
rcmd            sp4css10        SP4EN0          1
rcmd            sp4n10          SP4EN0          1
```

Look for the column labeled Key Version. If you find that the versions are different between the CWS and the nodes for the same service, for example rcmd, the `/usr/lpp/ssp/kerberos/etc/ext_srvtab` command can be used to create new server key files for each node.

The permissions are: -r-------- owner root and group system.

/etc/krb.conf

The SP authentication configuration file defines the local realm and the location of authentication servers for known realms.

```
# cat /etc/krb.conf
SP4EN0
SP4EN0 sp4en0 admin server
```

The permissions are: -rw-r--r-- owner root and group system.

/etc/krb.realms

This file contains an association of hostnames and authentication realms to specify the services provided by the host.

```
# cat /etc/krb.realms
sp4cw0.msc.itso.ibm.com SP4EN0
sp4n01.msc.itso.ibm.com SP4EN0
sp4sw01.msc.itso.ibm.com SP4EN0
sp4n05.msc.itso.ibm.com SP4EN0
sp4sw05.msc.itso.ibm.com SP4EN0
sp4n06.msc.itso.ibm.com SP4EN0
sp4sw06.msc.itso.ibm.com SP4EN0
sp4n07.msc.itso.ibm.com SP4EN0
sp4sw07.msc.itso.ibm.com SP4EN0
sp4n08.msc.itso.ibm.com SP4EN0
```

The permissions are: -rw-r--r-- owner root and group system.

/var/adm/SPlogs/kerberos

You must have the following two files in this directory:

- kerberos.log

  The permissions are: -rw-r--r-- owner root and group system.

- admin_server.syslog

  The permissions are: -rw-r--r-- owner root and group system.

These files contain messages about the behavior of the kerberos and kadmind daemons, respectively. You should check them to see if you have any error messages.

That is all you should verify with Kerberos.

3.4 General Environment

There are variables and files that are set when you install your SP, which are used to manage the system. Those variables include the characteristics of your nodes and CWS such as IP addresses, hostname, versions of AIX and PSSP installed, and others. You can quickly verify that these variables fit your expectations and that you have a consistent environment.

One command that is very helpful to do such verification is `splstdata`, with which you can display all the information of your environment that is taken from the SDR.

The main flags you can use for verification are:

-a Displays the following SDR LAN data only for nodes in the current system partition:

node_number

adapter_type

netaddr

netmask

host_name

type

rate

As shown in Figure 14, you will have all the IP addresses defined in the SDR for your SP, so do not worry if you do not see addresses for all the adapters that you have in your nodes. The basic ones that you should see will be the adapters for the service LAN and the switch adapters (if you have a switch, of course).

Check that the IP addresses and the netmasks are effectively correct.

```
List LAN Database Information

node#  adapt      netaddr          netmask       hostname  type t/r enet_ duplex
-------------------------------------------------------------------------------
    1   css0      192.168.14.1      255.255.255.0   sp4sw01  NA NA NA     NA       ""
    5   css0      192.168.14.5      255.255.255.0   sp4sw05  NA NA NA     NA       ""
    6   css0      192.168.14.6      255.255.255.0   sp4sw06  NA NA NA     NA       ""
    7   css0      192.168.14.7      255.255.255.0   sp4sw07  NA NA NA     NA       ""
    8   css0      192.168.14.8      255.255.255.0   sp4sw08  NA NA NA     NA       ""
    9   css0      192.168.14.9      255.255.255.0   sp4sw09  NA NA NA     NA       ""
   10   css0     192.168.14.10      255.255.255.0   sp4sw10  NA NA NA     NA       ""
   11   css0     192.168.14.11      255.255.255.0   sp4sw11  NA NA NA     NA       ""
   13   css0     192.168.14.13      255.255.255.0   sp4sw13  NA NA NA     NA       ""
   15   css0     192.168.14.15      255.255.255.0   sp4sw15  NA NA NA     NA       ""
    1    en0       192.168.4.1      255.255.255.0   sp4n01  bnc NA 10   half       ""
    5    en0       192.168.4.5      255.255.255.0   sp4n05  bnc NA 10   half       ""
    6    en0       192.168.4.6      255.255.255.0   sp4n06  bnc NA 10   half       ""
    7    en0       192.168.4.7      255.255.255.0   sp4n07  bnc NA 10   half       ""
    8    en0       192.168.4.8      255.255.255.0   sp4n08  bnc NA 10   half       ""
    9    en0       192.168.4.9      255.255.255.0   sp4n09  bnc NA 10   half       ""
   10    en0      192.168.4.10      255.255.255.0   sp4n10  bnc NA 10   half       ""
   11    en0      192.168.4.11      255.255.255.0   sp4n11  bnc NA 10   half       ""
   13    en0      192.168.4.13      255.255.255.0   sp4n13  bnc NA 10   half       ""
   15    en0      192.168.4.15      255.255.255.0   sp4n15  bnc NA 10   half       ""
```

Figure 14. Output of the splstdata -a Command

-b Displays the following SDR boot/install data:

node_number

host_name

hdw.enet.addr

boot_server

bootp_response

install_disk

last_inst_image

last_inst_time

next_inst_image

lppsource_name

pssp_version

This is one of the most helpful commands, as you can see in the following screen output. It gives you almost all the information you need to know to do maintenance tasks such as installation, configuration and upgrading of your nodes. The boot_server under the column labeled srvr is the installation server for that node: specifically, in this example, the installation server for all the nodes is node number zero (that means the CWS). If you have other installation servers, check that they are correct in the output of this command. You can also verify that the lppsource_name corresponds to the directory that contains the LPPS of the version of AIX that you want installed in that node, and that the version of PSSP listed is right. This verification is absolutely necessary before and after you do an installation or migration of any node.

```
 List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr      response            install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
---------------------------------------------------------------------------------
    1 sp4n01             02608C2E86CA     0      install                  hdisk0
                 default Fri_Apr__9_18:30:38             default          aix432
                PSSP-3.1              rootvg
    5 sp4n05             10005AFA0518     0      install           hdisk0,hdisk1
                 default             initial     bos.obj.ssp.432         default
                PSSP-3.1          alt_rootvg
    6 sp4n06             10005AFA17E3     0        diag                   hdisk0
                 default Fri_Apr__9_18:45:17             default          aix432
                PSSP-3.1              rootvg
    7 sp4n07             10005AFA1721     0        disk                   hdisk0
                 default Fri_Apr__9_18:51:19             default          aix432
                PSSP-3.1              rootvg
    8 sp4n08             10005AFA07DF     0        disk                   hdisk0
                 default Fri_Apr__9_18:53:33             default          aix432
                PSSP-3.1              rootvg
    9 sp4n09             0004AC4947E9     0        disk                   hdisk0
                 default Fri_Apr__9_18:26:04             default          aix432
                PSSP-3.1              rootvg
   10 sp4n10             0004AC494B40     0        disk                   hdisk0
                 default Fri_Apr__9_18:31:45             default          aix432
                PSSP-3.1              rootvg
   11 sp4n11             02608C2E7643     0        disk                   hdisk0
                 default Fri_Apr__9_18:43:10             default          aix432
                PSSP-3.1              rootvg
   13 sp4n13             02608C2E7C1E     0        disk                   hdisk0
                 default Fri_Apr__9_18:46:48             default          aix432
                PSSP-3.1              rootvg
   15 sp4n15             02608C2E78C9     0        disk                   hdisk0
                 default Sat_Apr_10_11:10:57             default          aix432
                PSSP-3.1              rootvg
```

A column that is specially important is the one labeled as response; it determines the answer that the CWS will have when you do a netboot of a node. If you are in your normal production environment, this field should be in status *disk*. If you find something different here, you may have a problem.

-e Displays SP object attributes and their values from the SDR.

There are some special fields you should pay attention to, such as the name and address of the CWS, the install_image, the code_version and the cw_lppsource_name.

The following screen output shows a typical response to this command.

```
# splstdata -e
List Site Environment Database Information

attribute               value
-----------------------------------------------------------------------------
control_workstation     sp4en0
cw_ipaddrs              9.12.0.4:192.168.4.140:
install_image           bos.obj.ssp.432
remove_image            false
primary_node            1
ntp_config              consensus
ntp_server              ""
ntp_version             3
amd_config              false
print_config            false
print_id                ""
usermgmt_config         true
passwd_file             /etc/passwd
passwd_file_loc         sp4en0
homedir_server          sp4en0
homedir_path            /home/sp4en0
filecoll_config         true
supman_uid              102
supfilesrv_port         8431
spacct_enable           true
spacct_actnode_thresh   80
spacct_excluse_enable   false
acct_master             0
cw_has_usr_clients      false
code_version            PSSP-3.1
layout_dir              ""
authent_server          ssp
backup_cw               ""
ipaddrs_bucw            ""
active_cw               ""
sec_master              ""
cds_server              ""
cell_name               ""
cw_lppsource_name       aix432
cw_dcehostname          ""
```

The install_image value is the default image that gets install. The code_version variable is default PSSP version which will be installed and the cw_lppsource_name is the level of NIM to install on the CWS.

There are other variables that you can also check, such as *amd_config,* which tells you if you have *amd* configured or not, and the *usermgmt_config* variable, which enables the user management option.

-n Displays the following SDR node data:

node_number

frame_number

slot_number

slots_used

initial_hostname

reliable_hostname

default_route

processor_type

processors_installed

description

```
 List Node Configuration Information

node# frame# slot# slots  initial_hostname  reliable_hostname  dcehostname
      default_route   processor_type processors_installed description
------------------------------------------------------------------------------
    1      1     1     4  sp4n01            sp4n01.msc.itso.  ""
        192.168.4.140           MP                    6 112_MHz_SMP_High
    5      1     5     1  sp4n05            sp4n05.msc.itso.  ""
        192.168.4.140           UP                    1 66_Mhz_PWR2_Thin
    6      1     6     1  sp4n06            sp4n06.msc.itso.  ""
        192.168.4.140           UP                    1 66_Mhz_PWR2_Thin
    7      1     7     1  sp4n07            sp4n07.msc.itso.  ""
        192.168.4.140           UP                    1 66_Mhz_PWR2_Thin
    8      1     8     1  sp4n08            sp4n08.msc.itso.  ""
        192.168.4.140           UP                    1 66_Mhz_PWR2_Thin
    9      1     9     1  sp4n09            sp4n09.msc.itso.  ""
        192.168.4.140           MP                    4 332_MHz_SMP_Thin
   10      1    10     1  sp4n10            sp4n10.msc.itso.  ""
        192.168.4.140           MP                    4 332_MHz_SMP_Thin
   11      1    11     2  sp4n11            sp4n11.msc.itso.  ""
        192.168.4.140           UP                    1 66_MHz_PWR2_Wide
   13      1    13     2  sp4n13            sp4n13.msc.itso.  ""
        192.168.4.140           UP                    1 66_MHz_PWR2_Wide
   15      1    15     2  sp4n15            sp4n15.msc.itso.  ""
        192.168.4.140           UP                    1 66_MHz_PWR2_Wide
```

The screen output shows information about all your nodes. Check that the hostnames are well-defined and that the default route is the one that you want to be (in a normal installation, you will see the IP address of the CWS as the default route).

There is another command that will help you to determine the status of your nodes in a specific moment in time, that is the `spmon` command, and the most complete report can be obtained using the `-d` and `-G` flags. In the output from this command you can verify that you have connection with the frame (or frames) and if you do, you can see the information listed in Figure 15 on page 84.

The fields that you should check are the following:

Verify that you can see all the frames and all the nodes in your SP. Check the column labeled Power to determine whether a particular node is listed as power off or on (remember that this power refers only to the *logical* power on, not the physical). If you have a node with this field set to no, you do not know if the node is logically powered off or physically powered off.

Verify that your Host Responds column is displaying `yes` for all nodes. If it is not, you may have a problem with your Ethernet TCP/IP connection; try to ping the node to see if it responds. If it does respond, problems within the High Available Infrastructure layer (specially hats) could be the cause of the missing host response. In this case refer to *RS/6000 SP: Problem Determination Guide,* SG24-4778*.*

Do the same check with the Switch Responds column. If you find a `no` there, you probably need to initialize your switch by executing the `Estart` command; Refer to 10.2, “Initialize the Switch” on page 228 for details.

The key switch works the same as on stand-alone RS/6000 workstations; it has three positions: normal, secure, and service. Verify that the key is in the position that you require. If you want to change the key to a different mode, you can execute the command `spmon -k` followed by the mode to which you want to change. For example, to change the key of node number 5 to service mode you execute:

```
# spmon -key service node05
```

The column Env Fail shows the status of your environment; this should *always* remain in value `no`. If this flag shows value `yes`, then you have a problem that must be resolved urgently because it indicates a failure in the system’s physical environment. For example, there could be a excessive temperature in the node because a fan is not working, or because the air conditioner is not working well.

The column of Front Panel LCD/LEDs will accomplish the same functions as in a stand-alone machine.

PSSP 2.4 introduced a new column called LCD/LED is Flashing that indicates if the displayed values are flashing (most times meaning hardware problems) or that the values are continuously changing.

```
# spmon -d -G
1.  Checking server process
    Process 13680 has accumulated 5 minutes and 45 seconds.
    Check ok

2.  Opening connection to server
    Connection opened
    Check ok

3.  Querying frame(s)
    1 frame(s)
    Check ok

4.  Checking frames

        Controller   Slot 17  Switch   Switch     Power supplies
Frame   Responds     Switch   Power    Clocking   A   B   C   D
----------------------------------------------------------------
  1       yes         yes      on         0       on  on  on  on
5.  Checking nodes
--------------------------------- Frame 1 ------------------------------------
Frame  Node    Node         Host/Switch  Key     Env   Front Panel   LCD/LED is
Slot   Number  Type  Power   Responds    Switch  Fail    LCD/LED      Flashing
-------------------------------------------------------------------------------
  1       1    high    on   yes   yes    normal   no   LCDs are blank   no
  5       5    thin    on   yes   yes    normal   no   LEDs are blank   no
  6       6    thin    on   yes   yes    normal   no   LEDs are blank   no
  7       7    thin    on   yes   yes    normal   no   LEDs are blank   no
  8       8    thin    on   yes   yes    normal   no   LEDs are blank   no
```

Figure 15. Output of the spmon -d -G Command

You should also verify that you have enough free space in your file systems, like / - and /var-filesystems. In particular, the /var filesystem should be checked because most of the system logfiles are written to this filesystem.

The services of high availability can also be verified; look into /var/ha/run/hats.<partition name> to search for a file called core or core.####. If you find one, that means that you had a problem with the topology service daemon and you can verify if the daemons are running by executing the command:

```
# lssrc -g hats
```

Alternatively, if you want detailed output you can list the information of the subsystem hats.<partition name> with the -l flag and it will look similar to the following:

```
# lssrc -l -s hats.sp4en0
Subsystem         Group            PID     Status
 hats.sp4en0      hats             20902   active
 Network Name   Indx Defd Mbrs St Adapter ID      Group ID
 SPether        [ 0]   11   11  S 192.168.4.140   192.168.4.140
 SPether        [ 0]              0x470e8685      0x470f7084
 HB Interval = 1 secs. Sensitivity = 4 missed beats
   2 locally connected Clients with PIDs:
 haemd( 15484) hagsd( 15234)
   Configuration Instance = 923696593
   Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
   CWS = 192.168.4.140
```

This command produces helpful information; for example, the State of the subsystem should be S (Stable). You can also verify that the number of nodes defined in the group of topology servers is equal to the number of Members active at a given moment; this number is the sum of the total nodes, plus your CWS, minus the nodes that are installed with PSSP version 1.2 or 2.1 that do not work with hats.

For more information about hats, refer to *RS/6000 SP High Availability Infrastructure,* SG24-4838*.*

3.5 System Data Repository (SDR)

As with the ODM in AIX, the SDR contains all the configuration information for the SP. It is located in the CWS and the nodes access it through the service network. The consistency of this database is needed to have your system working properly.

While there are commands that allow you to look at, modify, and delete your SDR, the recommendation is to not modify the SDR directly. Instead use the commands provided to administer the SP. However, if you find that you do need to modify the SDR, refer to 7.1.2, “Backing Up the SDR” on page 143 to learn how to back up the SDR before you do any modification. The SDR is located in the /spdata/sys1 directory and its subdirectory structure is shown in Figure 16 on page 86.

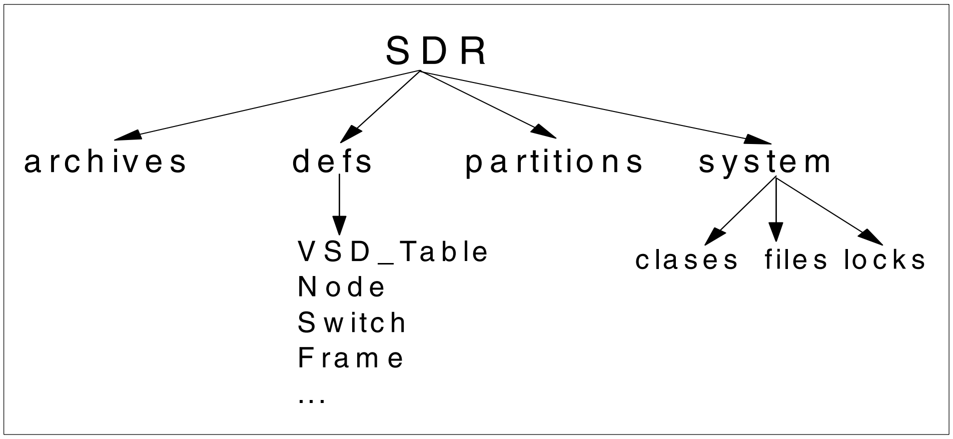


Figure 16. SDR Subdirectory Structure

The archives subdirectory could either be empty (if you have not taken any backup of the SDR), or it could contain the files of the backups that you have taken. In the defs subdirectory, you should have the header files for all the object classes. The partitions subdirectory should have at least one directory named as the IP address of the CWS, as well as other additional subdirectories for each additional partition that you have defined. The system subdirectory contains the classes and files which are global to the system.

To do a fast check of the SDR, issue the following command:

```
# SDR_test
SDR_test: Start SDR commandline verification test
SDR_test: Verification succeeded
```

If you have a different response, you have a problem with your SDR. In that case, do the following:

Check that the SDR daemons are running by typing:

```
# ps -ef | grep sdr
```

The response should be similar to:

```
root  2974 12024   1 16:46:02  pts/5  0:00 grep sdr
root 10850  7228   1 20:43:05      -  0:31 /usr/lpp/ssp/bin/sdrd 192.168.4.140
```

You will see one SDR daemon for each partition that you have in your system. You can also check this by executing the following command:

```
#lssrc -g sdr
```

The output of the command in this case is:

```
Subsystem         Group            PID     Status
 sdr.sp4en0       sdr              10850   active
```

Verify that the name of the subsystem is associated with the name of the partition.

You can also check if there is an entry in the inittab as follows:

```
sdrd:2:once:/usr/bin/startsrc -g sdr
```

However, the normal (and easiest) procedure if you have a problem with the SDR is to stop the subsystem and restart it; you can do this using the `stopsrc` and `startsrc` commands.

Chapter 4. Alternate and Mirrored rootvg

In this chapter we explain the new concepts of alternate and mirrored root volume groups which was introduced with PSSP 3.1. We will also look at the different boot modes which are required for doing software maintenance.

4.1 Alternate Boot System Image Function

There is a new feature in PSSP 3.1 that allows you to define, install and manage different, absolutely independent rootvg volume groups for an SP node or the CWS. However, only one rootvg volume group can be active at any time, and no changes are propagated from one rootvg to the other. This function is specially useful for fast migrations and testing of software before taking it into production.

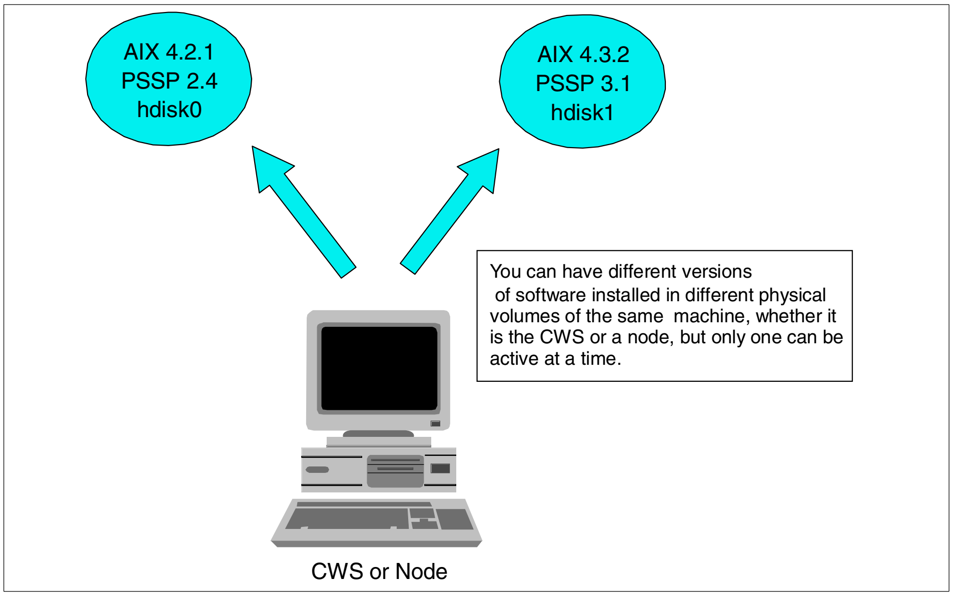

AIX 4.2.1
PSSP 2.4
hdisk0
AIX 4.3.2
PSSP 3.1
hdisk1
You can have different versions of software installed in different physical volumes of the same machine, whether it is the CWS or a node, but only one can be active at a time.
CWS or Node

Figure 17. Alternate rootvg

In order to support this new functionality a new class was added to the SDR: *Volume_Group* class.

The attributes of this new class are the following:

- *node_number:* The node number for Volume_Group.
- *vg_name*: The customer-supplied volume group name.

Default: rootvg.

- *pv_list*: A list of physical volumes.

  Default: hdisk0.

- *rvg*: Specifies whether the volume group is a root volume group.

  Valid values: true or false.

  Default: true.

- *quorum*: Specifies whether the quorum is set to on for this volume group.

  Valid values: true or false.

  Default: true.

- *copies*: Specifies the number of copies for the volume group.

  Valid values: 1or 2 or 3.

  Default: 1.

- *mapping*: Specifies whether mapping is set to on.

  Valid values: true or false.

  Default: false.

- *install_image*: The name of the mksysb install image to use for next install.

  Default: default.

- *code_version*: PSSP code version to use for next install.

  Default: the one defined in the environment.

- *lppsource_name*: Name of the lppsource resource to use for next install.

  Default: default.

- *boot_server*: The node_number of the boot/install server.

  Default: Default depends upon the node location.

- *last_install_image*: A string specifying the name of the last image installed on the node.

  Default: Initial.

- *last_bootdisk*: A string specifying the logical device name of the last volume from which the node booted.

  Default: Initial.

To manage this new class, five new commands were added to the SP system. These commands are:

- spmkvgobj
- spchvgobj
- sprmvgobj
- spmirrorvg
- spunmirrorvg

4.1.1 Prerequisites

To enable an alternate rootvg, you must have the following hardware:

- At least two hard disks (one for each rootvg volume group)
- A virtual battery on the SP nodes

The SP virtual battery is a power source that keeps NVRAM powered and the time of day clock running while the SP frame is plugged into a power outlet with power applied, or until a node is removed from the SP frame. Power is supplied by the node supervisor card in each node. This card provides line power even when the node is powered off.

On wide SP nodes, a virtual battery is installed from the factory. On thin nodes, a no charge Engineering Charge Action (ECA) can be ordered by your IBM representative; order ECA number ECA007. On the CWS, a battery is installed and no virtual battery ECA is required.

4.1.2 Defining an Alternate rootvg

In the following example, we have a thin node in slot 11 with two internal disks. We assume that AIX4.2.1 and PSSP 2.4 are already installed on the first physical disk (hdisk0). We now want to create an alternate rootvg on the second internal disk (hdisk1). AIX 4.3.2 and PSSP 3.1 will be installed on this disk.

We already copied the AIX 4.3.2 lppsources in the directory /spdata/sys1/install/aix432/lppsource, and the PSSP code in directory /spdata/sys1/install/pssplpp/PSSP-3.1.

In order to make a clear relationship between the installation device and the physical device, we recommend that you use the physical location code (00-00-0S-1,0) in the Physical Volume List field of the next input mask instead of using logical device names, like hdisk0. AIX assigns location device names dynamically to physical device names. Therefore it is not always obvious which physical device belongs to the logical device hdisk0.

To get the location code use the command:

```
# lsdev -Cc disk
```

The next screen shows a command output from one of our nodes.

```
# lsdev -Cc disk
hdisk0 Available 00-00-0S-0,0 2.0 GB SCSI Disk Drive
hdisk1 Available 00-00-0S-1,0 2.0 GB SCSI Disk Drive
```

As you can see, we have two internal scsi disk devices.

We want to define a alternate rootvg, *rootvg_432*. We can use the fast path SMIT command:

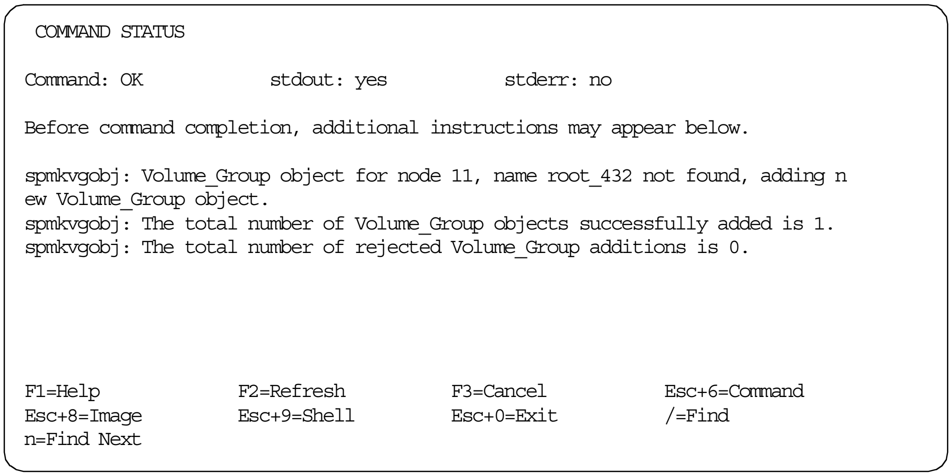
```
# smitty creatvg_dialog
```

```
 Create Volume Group Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Start Frame                                        [1]                      #
  Start Slot                                         [11]                     #
  Node Count                                         [1]                      #

  OR

  Node List                                          []

  Volume Group Name                                  [rootvg_432]
  Physical Volume List                               [00-00-0S-1,0]
  Number of Copies of Volume Group                    1                      +
  Boot/Install Server Node                           []                       #
  Network Install Image Name                         []
  LPP Source Name                                    [aix432]
  PSSP Code Version                                   PSSP-3.1               +
  Set Quorum on the Node                                                     +



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

The appropriate command is:

```
# spmkvgobj -r root_432 -h ’00-00-0S-1,0’ -i bos.obj.ssp.432 \
  -v aix432 -p PSSP-3.1 -l 11
```

The following screen output shows that the alternate rootvg *root_432* did not exist and was therefore created.



```
 COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

spmkvgobj: Volume_Group object for node 11, name root_432 not found, adding n
ew Volume_Group object.
spmkvgobj: The total number of Volume_Group objects successfully added is 1.
spmkvgobj: The total number of rejected Volume_Group additions is 0.




F1=Help             F2=Refresh          F3=Cancel           Esc+6=Command
Esc+8=Image         Esc+9=Shell         Esc+0=Exit          /=Find
n=Find Next
```

4.1.3 Installing an Alternate rootvg

If we want to install this alternate rootvg we need to set the bootp_response attribute for the node to *install*. Do this with the following steps:

1. Use the command: `smitty server_dialog`

   Fill all the input fields as shown in the screen. Also set the Run Setup server option to no, for the time being, so that you can check the SDR.

```
 Boot/Install Server Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Start Frame                                        [1]                     #
  Start Slot                                         [11]                    #
  Node Count                                         [1]                     #

  OR

  Node List                                          []

  Response from Server to bootp Request               install               +
  Volume Group Name                                  [rootvg_432]
  Run setup_server?                                   no                    +




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

The appropriate command is:

```
# spbootins -l 11 -r install -c rootvg_432
```

2. Check the SDR by using the command `splstdata -b`; this should reflect the new image.

```
# splstdata -b -l 11

List Node Boot/Install Information

node#        hostname  hdw_enet_addr srvr     response           install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
-------------------------------------------------------------------------------
   11 sp4n11            02608C2E7643    0      install           00-00-0S-1,0
                 default             initial         default            aix432
                PSSP-3.1         rootvg_432
```

3. Now you can either run `setup_server` to initiate an install operation, or you can do it directly by using the wrapper-commands. Execute the following commands to prepare for the installation of node sp4n11:

```
# setup_server
```

or

```
# Efence -G -autojoin sp4n11
# create_krb_files
# mkconfig
# mkinstall
# export_clients
# allnimres -l 11
```

After the node comes up, the `lspv` command in Figure 18 on page 95 shows that the rootvg is located on hdisk1 and that hdisk0 is unassigned; even if our first root volume group is installed on this disk.

```
# lspv
hdisk0         0000097330a4837d    None
hdisk1         00000973967bdfa2    rootvg
hdisk2         000003411ee207da    None
hdisk3         00003550d567d4e3    None
```

Figure 18. lspv Command Output

4.1.4 Switching between Alternate System Images

Now that we have at least two root volume groups, how can we check this and how can we switch between these two system images?

To verify that we have more than one root volume group definition for node 11 we run the `splstdata` command:

```
# splstdata -v -l 11
                 List Volume Group Information

node# name           boot_server quorum copies   code_version lppsource_name
        last_install_image       last_install_time   last_bootdisk
        pv_list
-----------------------------------------------------------------------------
   11 rootvg           0          true      1         PSSP-2.4 aix421

        default                  Fri_Apr__9_18:43:10_EDT_1999 hdisk0

        hdisk0
   11 rootvg_432       0          true      1         PSSP-3.1 aix432

        default                  initial            hdisk0
```

As you can see, the new `-v` option for the splstdata command shows all defined volume groups for a node.

The next question is, how do we know which root volume group will be used for the next boot? We use `splstdata -b` to display this information:

```
# splstdata -b -l 11
                  List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response            install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
-------------------------------------------------------------------------------
   11 sp4n11             02608C2E7643    0      disk                     hdisk0
                 default Fri_Apr__9_18:43:10            default         aix421
                PSSP-2.4         rootvg
```

As we can see, the current active system is the AIX 4.2.1 version on hdisk0.

If we want to switch to an alternate root volume group we will use the `spbootins` command:

```
# spbootins -c rootvg_432 -l 11 -s no
```

The rootvg_432 becomes the current root volume group for subsequent installation and customization. Let us check, using the `splstdata -b` command, that our changes are carried out.

```
# splstdata -b -l 11

List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response            install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
-------------------------------------------------------------------------------
   11 sp4n11             02608C2E7643    0      disk                     hdisk1
                 default            initial          default             aix432
                PSSP-3.1         rootvg_432
```

There is one final step missing. We have to modify the bootlist of node 11 so that the next boot will be done using hdisk1 (rootvg_432) instead of hdisk0 (rootvg). We do this by using the new PSSP command `spbootlist`.

Like the normal AIX bootlist command, the `spbootlist` command will look at the *vg_name* attribute in the *Volume_Group* object, determine which physical volume(s) are in the volume group, and set the bootlist to them. Let us look at an example:

```
# spbootlist -l 11
```

This command takes the information stored in the SDR and issues a remote command on the selected node. Now the next boot will bring up the alternate root volume group *rootvg_432* on node 11.

4.1.5 Alternate rootvg Highlights

No matter what name you give to the volume group object in the SDR on the CWS, the name of the active volume group of the node will be *rootvg.*

Only one rootvg can be active at any time.

> **Important**
>
> When you boot your node from one rootvg on one specific physical volume, you will see that the other physical volumes assigned to an alternate rootvg appear as unassigned (see Figure 18 on page 95). Do not define anything on those physical volumes or you will destroy your alternate rootvg.

The normal bootlist within your nodes will be modified by using the `spbootlist` command. You should also think about modifying your service bootlist. Why? Imagine that you are trying to boot your system in service mode to do some diagnostic work and your system is not coming up. If you have defined the second physical disk (with the alternate rootvg installed on it) as an alternate boot device, your system will try to boot from this device. Your node probably will boot from this device. In this case your node is still accessible.

4.2 Rootvg Mirroring

PSSP 3.1 supports mirroring of the root volume group. Rootvg Mirroring helps you achieve AIX High Availability. You can have multiple copies of your operating system spread over multiple physical volumes. If one physical volume fails, you still have other valid copies and your system remains operational.

You can configure two or three copies of each logical volume of the operating system (the original plus one or two copies). The only logical volume that cannot be mirrored is the dump logical device.

When we talk about mirroring we need to think about disk quorum. When quorum is enabled, a voting scheme will be used to determine if the number of physical volumes that are up is enough to maintain a quorum. If quorum is lost, the entire volume group will be taken offline to preserve data integrity. If quorum is disabled, the volume group will remain on line as long as there is at least one valid physical volume. We chose disk quorum to be enabled, which is generally safer.

4.2.1 Mirroring Example

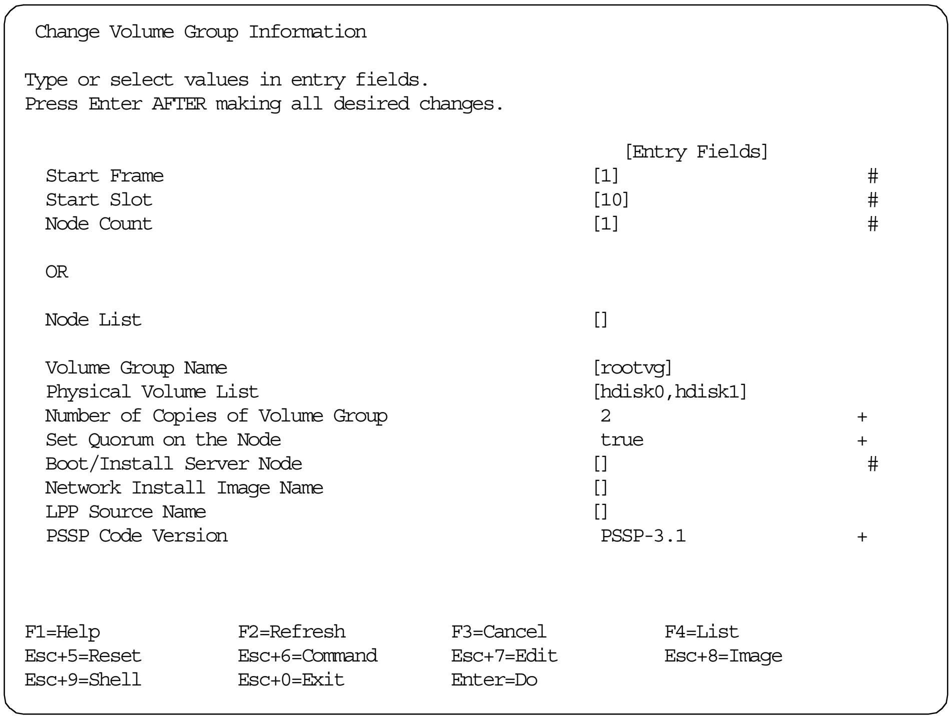
We will install node 10 with two copies of the operating system.

Enter the information about the node so that your volume group is defined with two copies. You can use the `spchvgobj` command to modify the information about your node. You can execute:

```
# spchvgobj -r rootvg -h hdisk0,hdisk1 -c 2 -l 10
```

If you use SMIT, type:

```
# smitty changevg_dialog
```



```
 Change Volume Group Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
  Start Frame                                         [1]                    #
  Start Slot                                          [10]                   #
  Node Count                                          [1]                    #

  OR

  Node List                                           []

  Volume Group Name                                   [rootvg]
  Physical Volume List                                [hdisk0,hdisk1]
  Number of Copies of Volume Group                     2                    +
  Set Quorum on the Node                               true                 +
  Boot/Install Server Node                            []                     #
  Network Install Image Name                          []
  LPP Source Name                                     []
  PSSP Code Version                                    PSSP-3.1             +



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

With these instructions, we specify that the rootvg volume group will be installed on hdisk0 and hdisk1. AIX takes care that each copy goes to a separate physical volume. Recall that the prerequisite to mirroring is to have at least two physical volumes in the volume group of interest. You can define mirroring at installation time or dynamically on a running node.

Now we can proceed with our normal installation. Afterwards, you can verify that the logical volumes are mirrored into the rootvg by executing the following command:

```
# lsvg -l rootvg
```

Check that the number of physical partitions dedicated to a logical volume is twice the number of logical partitions for a mirrored rootvg with two instances. The only exception is the sysdump logical volume, which cannot be mirrored.

You can initiate the mirroring of the root volume group on a running system by issuing the following command:

```
# spmirrorvg -l 10 -f
```

This assumes that you defined the mirroring of the rootvg already, as described before. The command uses information found in the Volume_Group object to initiate the mirroring.

4.2.2 Unmirroring a Root Volume Group

If you decide that you do not want the rootvg mirrored anymore, you can modify the definition in the SDR and customize the node. The following commands are required to unmirror node 10.

1. Modify the information for the volume group:

   ```
   # spchvgobj -r rootvg -c 1 -l 10
   ```

2. Change the bootp response for that node and run `setup_server` in the CWS:

   ```
   # spbootins -r customize -l 10
   ```

3. Reboot the node:

   ```
   # cshutdown -r -N 10
   ```

You could also remove the mirroring by using the normal AIX commands on the node or through SMIT, and modify the information in the SDR afterwards. Otherwise, the next time you customize that node, the system will try to recreate the mirroring.

4.3 Boot Modes

The SP nodes can be booted up to several system modes. You can choose among five different modes:

- disk
- install
- migrate
- customize
- diag

You can select the mode that you want through the SMIT panels or by using the spbootins command. This mode will determine the actions taken when the setup_server script is executed. It will influence the next boot procedure.

If you want to check the current boot mode settings for a node, use the splstdata -b command and look for the column labeled “response”. The following is an example of the output of this command:

```
 List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response           install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
-------------------------------------------------------------------------------
    1 sp4n01             02608C2E86CA    0      disk                    hdisk0
                 default Fri_Apr__9_18:30:38             default         aix432
                PSSP-3.1                rootvg
    5 sp4n05             10005AFA0518    0      install          hdisk0,hdisk1
                 default             initial     bos.obj.ssp.432        default
```

To modify the bootp_response attribute of a node and save this in the SDR, type the following command:

```
# spbootins -r <mode> -l <node list>
```

When you run this command, the default option is to also run the setup_server script.

The following sections explain briefly the procedures which are executed when you change the boot mode.

4.3.1 Disk Mode

We can say that this is the "normal" mode for your SP system, meaning that you should have your nodes in this mode while they are in the production environment.

In this mode, when you run `setup_server`, all the resource allocations are removed for this NIM client. The file /tftpboot/<hostname>.info is also removed, as well as the entry in the /etc/bootptab for that client. All this means that the bootp request will be ignored and that your node will boot from the local disk.

You can reboot the node as if it were a normal standalone machine. It will boot from the local disk and start the operating system (if the key is in normal) or the diagnostics menu (if the key is in service).

4.3.2 Install Mode

You use this mode when you want to completely install a node. When `setup_server` is run, the following procedures are executed:

- Allocate SPOT
- Allocate lpp_source
- Allocate bosinst_data
- Allocate psspscript
- Allocate mksysb
- Update /etc/inetd.conf
- Create <reliable_hostname>.install_info
- Create <reliable_hostname>.config_info
- Create the Kerberos file <reliable_hostname>-new_srvtab

Then, in order to install your node, you must boot from the network.

4.3.3 Migrate Mode

This mode indicates that you want the server to perform a migration on the specified nodes. The following procedures take place when running the server in this mode:

- Allocate SPOT
- Allocate lpp_source
- Allocate bosinst_data
- Allocate psspscript
- Update /etc/inetd.conf
- Create <reliable_hostname>.install_info
- Create <reliable_hostname>.config_info

- Create the Kerberos file <reliable_hostname>-new_srvtab

4.3.4 Customize Mode

In this mode the setup_server script deallocates all resources of a given NIM client, removes the NIM client, and recreates it. The files:

- <reliable_hostname>.install_info
- <reliable_hostname>.config_info
- <reliable_hostname>-new-srvtab

are created under the /tftpboot directory.

4.3.5 Diag Mode

This mode allows you to bring up your node in diagnostic mode. You have to reboot it.

The `setup_server` script does the following:

- Allocate SPOT
- Allocate bosinst_data
- Update /etc/inetd.conf
- Create <reliable_hostname>.install_info
- Create <reliable_hostname>.config_info

The difference between booting a node locally with the key in service position and booting from the network in diag mode is that in the first case, you are still booting from your local disk and loading the diagnostic software from that disk, while in the second case, the diagnostic software will be mounted from the CWS to the node.

4.3.6 Maintenance Mode

This mode is useful when you encounter boot problems. You will get a maintenance menu, from which you can initiate several actions. This mode also requires a netboot.

The `setup_server` script does the following:

- Allocate SPOT
- Allocate lpp_source
- Allocate bosinst_data
- Update /etc/inetd.conf
- Create <reliable_hostname>.install_info
- Create <reliable_hostname>.config_info

Chapter 5. Updating Your System

This chapter is to be used for applying Program Temporary Fixes (PTFs) for AIX, Parallel System Support Program (PSSP) and other Licensed Program Products (LPPs) in SP. If you are planning for migration of the AIX version or PSSP software, refer to Chapter 6, “Migrating PSSP and AIX to Later Versions” on page 123.

Software corrections or enhancements are released in the form of PTFs. These patches can be ordered through the IBM Software Support Center, or they can be downloaded either from the Web or by using standalone FixDist servers. For more information on downloading PTFs, refer to Appendix A, “Downloading PTFs for RS/6000 and SP” on page 243.

When installing PTFs on a production system, you must first take certain precautionary measures. In this chapter we show the steps for successfully installing PTFs on the CWS and the nodes.

Before applying any PTFs, you should review the memos, the so-called README files. These README files are most times related to filesets and include the latest information, which is sometimes not included in the manuals. They tell you what problems are being fixed and whether any specific prerequisite filesets are required. They also tell you whether a shutdown of the node is required for the fixes to take effect. Generally, it is suggested you install the fixes in apply mode so that if there is any problem, you can reject the filesets that have been applied. In some cases we suggest you commit the fixes, as they may be mandatory. For these reasons, it is imperative that you read the memos before proceeding.

The README files for all PSSP versions and their LPPs are available at:
`http://www.rs6000.ibm.com/support/sp/sp_secure/readme/`

As an example, when you apply the PTF ssp.css.2.4.0.2, you get the following output on the console:

NOTE: IX78007 and IX78629

PTF ssp.css 2.4.0.2 contains a fix to fault_service and fault_service_SP. In order for this fix to take effect, you must reboot the node. If you are not encountering this switch problem, you can wait until your next scheduled reboot before applying this PTF.

5.1 Applying PTFs

Whenever you apply PTFs for PSSP software, you must first install them on the CWS and verify that your SP-related environments are working OK. Refer to Chapter 3, “Verifying Your SP” on page 67 for instructions on checking the SP setup. After the CWS is found to be OK, you must next plan to choose one node for applying the PTF and test it for proper operation.

For installing on the nodes, two approaches are documented in the *Installation and Migration Guide*. The first approach is to apply the updates on a per node basis using `installp` , and the second approach is to reinstall the node using a customized mksysb image with the PTFs installed.

Generally in a production environment, a lot of customization will have been done, and shell scripts will have been written for performing various day-to-day administrative tasks. So the customer would most likely choose to install the PTFs in each node rather than reinstalling the nodes with a new image.

Now let us look at the steps to be followed for installing the PTFs on the CWS and the nodes.

5.2 Applying AIX PTFs in the CWS

The steps for applying the PTFs are as follows:

1. Create a mksysb backup image of the CWS. We recommend that you always take two backups, on two different tapes, and never trust just one tape. Check that the tape is OK by listing the contents of the tape using the command `smitty lsmksysb`. For more information on taking a mksysb backup, refer to 7.1.1, “Mksysb and Savevg” on page 139.
2. Copy the PTFs to the lppsource directory /spdata/sys1/install/<aix_version>/lppsource.
3. Create a new .toc file by executing the command:

   ```
   # nim -o check <lpp_source>
   ```

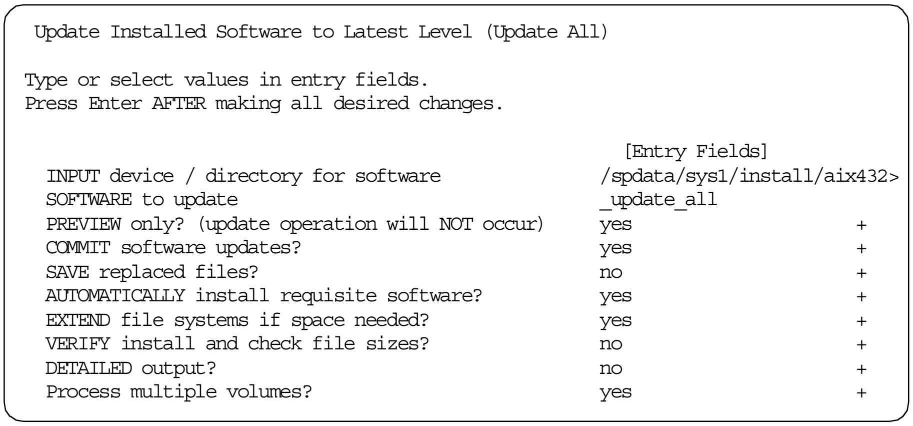
   or

   ```
   # inutoc /spdata/sys1/install/aix432/lppsource
   ```

4. Update the new PTFs to the CWS using SMIT as follows:

   ```
   # smitty update_all
   ```

   Input device for directory/software: `/spdata/sys1/install/aix432/lppsource`



```
 Update Installed Software to Latest Level (Update All)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  INPUT device / directory for software              /spdata/sys1/install/aix432>
  SOFTWARE to update                                 _update_all
  PREVIEW only? (update operation will NOT occur)    yes                    +
  COMMIT software updates?                           yes                    +
  SAVE replaced files?                               no                     +
  AUTOMATICALLY install requisite software?          yes                    +
  EXTEND file systems if space needed?               yes                    +
  VERIFY install and check file sizes?               no                     +
  DETAILED output?                                   no                     +
  Process multiple volumes?                          yes                    +
```

First run with the PREVIEW only option set to yes, and check that the prerequisites are available. If it is OK, then continue installing the PTFs by changing the PREVIEW only option to no.

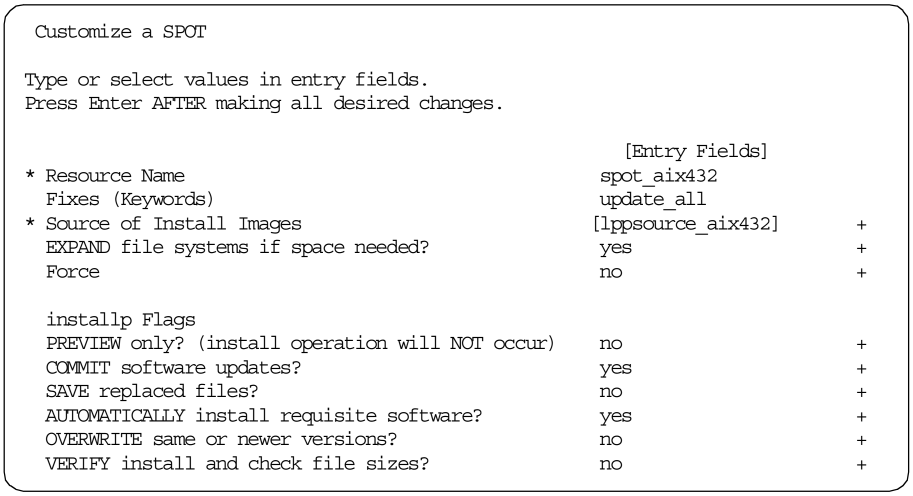
During the installation, check for any errors being reported. It is also possible to check this by viewing the smit.log file any time after the installation is over.

5. Update the SPOT with the PTFs in the lppsource directory using the following command, as shown in the next screen:

   `# smitty nim_res_op`

   Resource name: `spot_aix432`.

   Operation to perform: `update_all`



```
 Customize a SPOT

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Resource Name                                   spot_aix432
  Fixes (Keywords)                                update_all
* Source of Install Images                       [lppsource_aix432]      +
  EXPAND file systems if space needed?            yes                    +
  Force                                           no                     +

  installp Flags
  PREVIEW only? (install operation will NOT occur)    no                 +
  COMMIT software updates?                        yes                    +
  SAVE replaced files?                            no                     +
  AUTOMATICALLY install requisite software?       yes                    +
  OVERWRITE same or newer versions?               no                     +
  VERIFY install and check file sizes?            no                     +
```

6. If the status of the install is OK, then you are through with the update of the AIX PTFs in the CWS. If the status of the install is FAILED, then you should review the output for the cause of the failure and resolve the problem.

5.3 Applying AIX PTFs on the Nodes

Applying PTFs on the nodes can be done by three different methods. Before applying the PTFs, take a mksysb backup of the node. For more information on taking a mksysb backup, refer to 7.1.1, “Mksysb and Savevg” on page 139.

The three different methods for installing AIX PTFs are as follows:

1. Creating a new image which contains all the PTFs in one node and propagating the image to the nodes.
2. Mounting the lppsource directory from the CWS to the nodes, and installing the PTFs manually using `smitty update_all` for all nodes.
3. If you have the DSMIT package installed, then you can install using `dsmitty update_all` to all the nodes in one effort.

The following sections details these methods.

5.3.1 Method 1: Applying AIX PTFs Using mksysb Install Method

To install PTFs using a new image, double check to be sure that you have not done any customization specific to the nodes. If you have done so, you must not use this method of installation.

This method of updating PTFs is suggested only when all the nodes are identical as far as the configuration and LPPs are concerned.

The steps for updating PTFs using this method are:

1. Choose one node for installing the PTFs and testing the image. In the lab, we chose the node sp2n10 for this purpose.
2. Login as root on the node and mount the lppsource directory of the CWS on /mnt using the command:

   ```
   # mount sp2en0:/spdata/sys1/install/aix432/lppsource /mnt
   ```

   If you get any errors, check whether the directory is NFS-exported by looking into the file /etc/exports.
3. Update the PTFs using the command:

   ```
   # smitty update_all
   ```

   Input device for directory/software: `/mnt`.

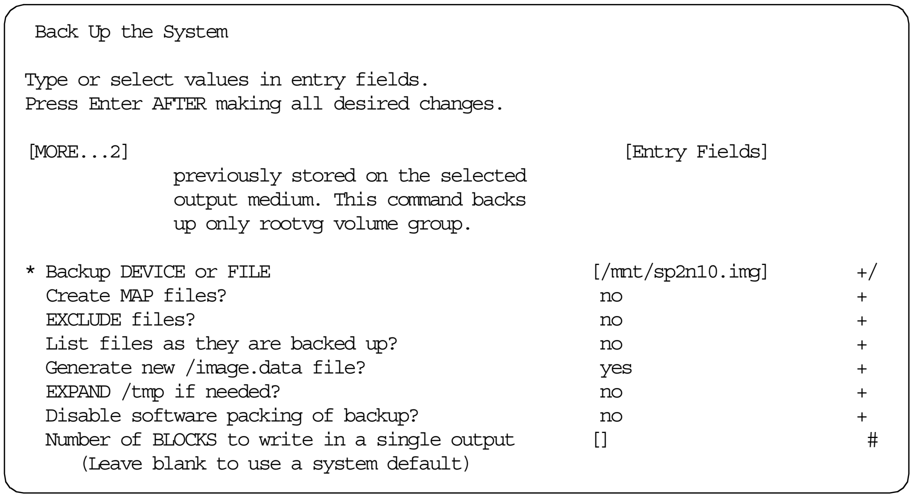
   First run with the PREVIEW only option set to yes, and check that all the prerequisites are available. If it is OK, continue installing the PTFs by changing the PREVIEW only option to `no`.

   Check for any errors being reported during the installation. You can also check this by viewing the smit.log file any time after the installation is over.
4. Create the `mksysb` image from this node. It is always advisable to have the image kept in the /spdata/sys1/install/images/<new_image_file>. To create the image of node sp2n10 on the CWS, first export the directory /spdata/sys1/install/images to node sp2n10. For example, to export /spdata/sys1/install/images from sp2en0(CWS) to sp2n10 issue the following command from the CWS:

   ```
   # mknfsexp -d '/spdata/sys1/install/images' -t 'rw' -r 'sp2n10' '-B'
   ```
5. Mount the directory locally on node sp2n10 using the command:

   ```
   # mount sp2en0:/spdata/sys1/install/images /mnt
   ```
6. Create the mksysb image using `smitty mksysb`.

   This is shown in the following screen.



```
 Back Up the System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...2]                                              [Entry Fields]
              previously stored on the selected
              output medium. This command backs
              up only rootvg volume group.

* Backup DEVICE or FILE                              [/mnt/sp2n10.img]        +/
  Create MAP files?                                   no                      +
  EXCLUDE files?                                      no                      +
  List files as they are backed up?                   no                      +
  Generate new /image.data file?                      yes                     +
  EXPAND /tmp if needed?                              no                      +
  Disable software packing of backup?                 no                      +
  Number of BLOCKS to write in a single output       []                       #
     (Leave blank to use a system default)
```

7. Unmount the /mnt filesystem on the node and remove the NFS export of the directory /spdata/sys1/install/images from the CWS. If you have the Switch installed, fence the node using the command `Efence sp2n10`. Reboot the node for the PTFs to take effect.

8. After the node has been rebooted, unfence the Switch using the command:

   ```
   # Eunfence sp2n10
   ```

   Now the image is ready on the CWS for installation on the other nodes. However, you should first check that all the operations on this node are working without any problems. It is generally advisable to keep this node under observation for a few days before installing the image on the other nodes.

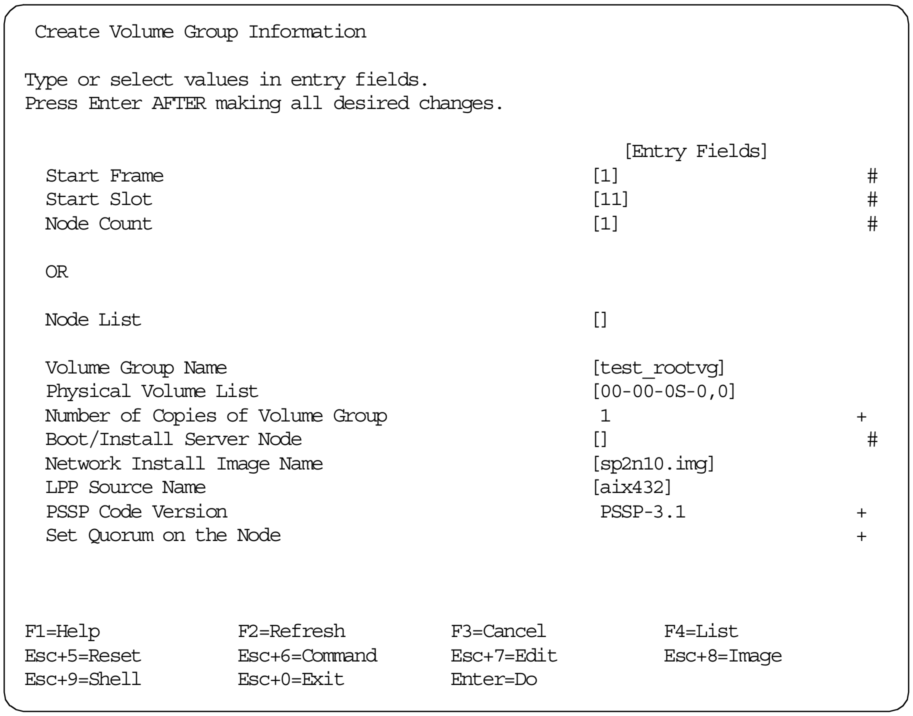
Propagating the image to the node:

To test the image, we will install our new image on node sp2n11. The following steps install node sp2n11 using the new image sp2n10.img which we created in step 6 of 5.3.1, “Method 1: Applying AIX PTFs Using mksysb Install Method” on page 106.

1. For our test installation we define an alternate rootvg as described in 4.1.2, “Defining an Alternate rootvg” on page 91. Execute the command:

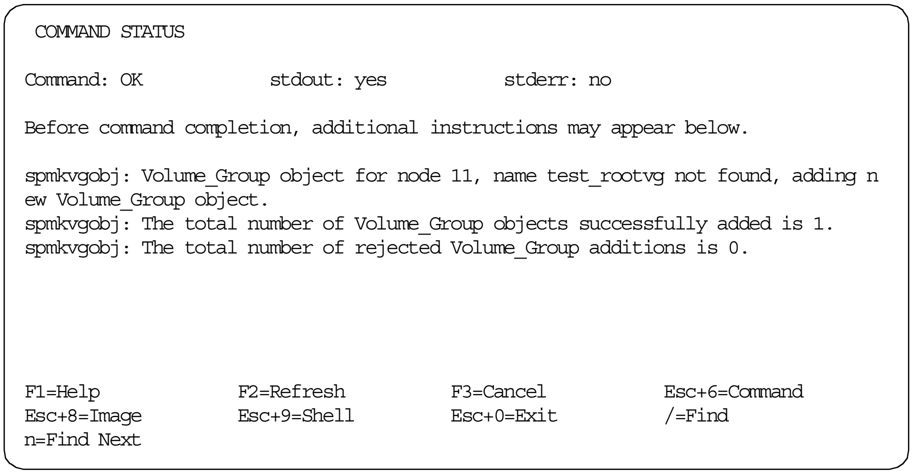
   ```
   # smitty createvg_dialog
   ```

   Fill in the input fields as shown in the following screen. The only difference here, when compared to the normal setup, is that the Network Install Image Name will be set to sp2n10.img.



```
 Create Volume Group Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
  Start Frame                                         [1]                      #
  Start Slot                                          [11]                     #
  Node Count                                          [1]                      #

  OR

  Node List                                           []

  Volume Group Name                                   [test_rootvg]
  Physical Volume List                                [00-00-0S-0,0]
  Number of Copies of Volume Group                     1                      +
  Boot/Install Server Node                            []                       #
  Network Install Image Name                          [sp2n10.img]
  LPP Source Name                                     [aix432]
  PSSP Code Version                                    PSSP-3.1               +
  Set Quorum on the Node                                                      +



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

The following screen output shows that the alternate rootvg *test_rootvg* did not exist and was therefore created.

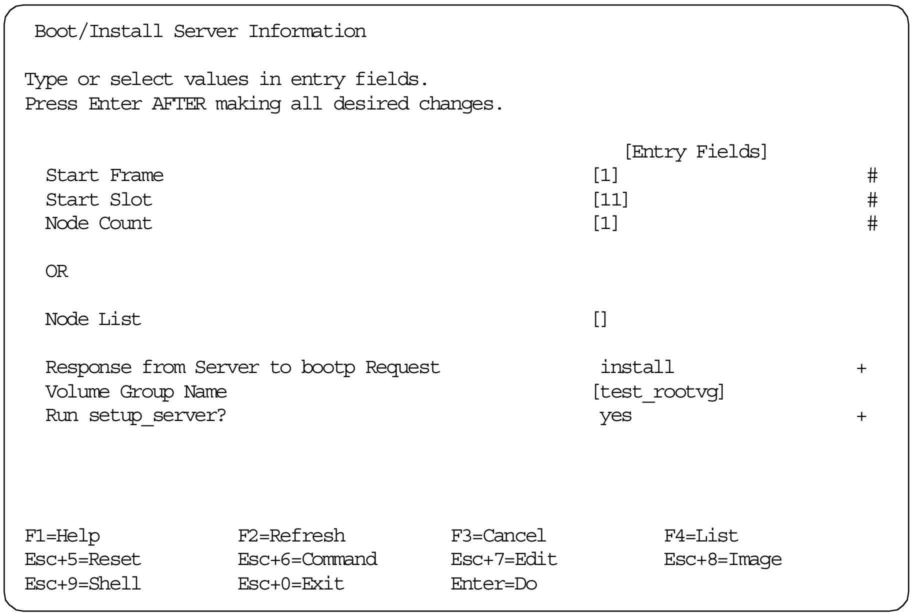


```
 COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

spmkvgobj: Volume_Group object for node 11, name test_rootvg not found, adding n
ew Volume_Group object.
spmkvgobj: The total number of Volume_Group objects successfully added is 1.
spmkvgobj: The total number of rejected Volume_Group additions is 0.




F1=Help             F2=Refresh          F3=Cancel           Esc+6=Command
Esc+8=Image         Esc+9=Shell         Esc+0=Exit          /=Find
n=Find Next
```

2. The next step is to set the bootp_response attribute for the node to *install*. Use the command:

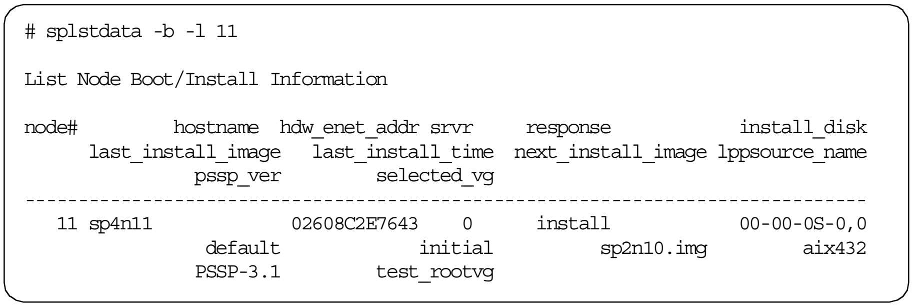
```
# smitty server_dialog
```

3. Fill all the input fields as shown in the screen. Also set the Run Setup server option to no, for the time being, so that you can check the SDR.



```
 Boot/Install Server Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
  Start Frame                                     [1]                    #
  Start Slot                                      [11]                   #
  Node Count                                      [1]                    #

  OR

  Node List                                       []

  Response from Server to bootp Request            install              +
  Volume Group Name                               [test_rootvg]
  Run setup_server?                                yes                  +



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

4. Check the SDR by using the command `splstdata -b`; this should reflect the new image.



```
# splstdata -b -l 11

List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response           install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
-------------------------------------------------------------------------------
   11 sp4n11            02608C2E7643    0      install            00-00-0S-0,0
                 default             initial          sp2n10.img        aix432
                PSSP-3.1         test_rootvg
```

5. Now you can either run `setup_server` to initiate an install operation, or you can do it manually using wrappers only for this node. For doing the manual steps to install the node, execute the following commands to prepare for the installation of node sp2n10:

```
# Efence sp2n10
```

```
# create_krb_files
# mkconfig
# mkinstall
# mknimres -l 0
# export_clients
# allnimres -l 10
```

When you execute these commands, the screen output will look like the following:

```
# Efence sp2n10
All nodes successfully fenced.

# create_krb_files
create_krb_files: tftpaccess.ctl file and client srvtab files created/updated
on server node 0.
# mkconfig

# mkinstall

# mknimres -l 0
mknimres: Copying /usr/lpp/ssp/install/bin/pssp_script to /spdata/sys1/install/p
ssp/pssp_script.
mknimres: Copying /usr/lpp/ssp/install/config/bosinst_data_prompt.template to /s
pdata/sys1/install/pssp/bosinst_data_prompt.
mknimres: Copying /usr/lpp/ssp/install/config/bosinst_data_migrate.template to /
spdata/sys1/install/pssp/bosinst_data_migrate.
mknimres: Successfully created the mksysb resource named mksysb_2 from dir
 /spdata/sys1/install/images/sp2n10.img on sp2en0.

# export_clients
export_clients: File systems exported to clients from server node 0.

# allnimres -l 10
exportfs: /spdata/sys1/install/images/sp2n10.img: parent-directory (/spdata/sys1
/install/images) already exported
exportfs: /spdata/sys1/install/images/sp2n10.img: parent-directory (/spdata/sys1
/install/images) already exported
allnimres: Node 10 (sp2n10) prepared for operation: install.
```

6. Start the netboot and see that the node is getting installed using this new image.

5.3.2 Method 2: Applying AIX PTFs Using SMIT or dsh

This method is used for installing the PTFs on the nodes by using SMIT or `dsh` commands. For installing on the nodes using SMIT, you login to each node and apply the PTFs. It is also possible to install the PTFs on the nodes from the CWS by using the `dsh` command.

As previously mentioned, for any of these options, it is better to install the PTFs in one node and do the testing there before applying them to all the nodes. In the lab, we selected sp2n10 for installing the PTFs and testing the node. Here we discuss both options for installing the PTFs.

Option 1: Using SMIT

1. Login to sp2n10 as root.
2. Mount the lppsource directory of the CWS in sp2n10 using the command:

   ```
   # mount sp2en0:/spdata/sys1/install/aix432/lppsource /mnt
   ```

3. Apply the PTFs using the command:

   ```
   # smitty update_all
   ```

   Input device for directory/software: `/spdata/sys1/install/aix432/lppsource`

   First run with the PREVIEW only option set to `yes`, and check that all prerequisites are met. If it is OK, install the PTFs with the PREVIEW only option set to `no`.

   While the installation is running, check for any errors being reported. You can also check this by viewing the smit.log file any time after the installation is complete.

4. Unmount the directory which you had mounted in step 2 by using the command:

   ```
   # umount /mnt
   ```

   While installing the PTFs, if you get any output saying that a reboot is required for the PTFs to take effect, you can do a reboot of the node.

5. Before initiating a reboot, if you have a switch, you will have to fence it using the command:

   ```
   # Efence sp2n10
   ```

6. After the system has been rebooted, unfence the node using the command:

   ```
   # Eunfence sp2n10
   ```

   Now it is time to test the node for proper operation. Once all tests are OK, you can install the other nodes. To install the other nodes individually using SMIT, follow the same steps.

Option 2: Using dsh

This method is used for installing the PTFs on the nodes by using the `dsh` command from the CWS.

1. Select sp2n10 as the working collective member for executing the commands by using dsh:

   ```
   # dsh -w sp2n10
   ```

2. Mount the lppsource directory of the CWS to node sp2n10 by using the command:

   ```
   # dsh> mount sp2en0:/spdata/sys1/install/aix432/lppsource /mnt
   ```

3. Install the PTFs by using the command:

   ```
   # dsh>/usr/lib/instl/sm_inst installp_cmd -a -d '/mnt' -f '_update_all'
   '-pcNgX'
   ```

   This command with the p option is for preview, and with the c option is for commit. For installing, give the command without the p option. If you also do not want to commit, remove the c option.

4. Unmount the directory that you mounted in step 2 by using the command:

   ```
   # umount /mnt
   ```

   While installing the PTFs, if you get any output saying that a reboot is required for the PTFs to take effect, it is better to reboot the node.

5. Before initiating a reboot, if you have a Switch, you will have to fence it by using the command:

   ```
   # Efence sp2n10
   ```

6. After the system has been rebooted, unfence the node by using the command:

   ```
   # Eunfence sp2n10
   ```

For installing to all nodes, use the `dsh -a` option to have the nodes in the working collection. If you want only selected nodes, use the command `dsh -w sp2n01,sp2n05,sp2n06,sp2n07` to install only in these nodes.

5.3.3 Method 3: Applying AIX PTFs Using DSMIT

Distributed System Management Interface Tool (DSMIT) adds functionality to SMIT by allowing the SMIT interface to build commands for system management and distribute them to other clients on a network. DSMIT does not use the .rhosts file for executing the command on the clients, and therefore it does not create any security exposure. DSMIT security is based on well-established crypto routines and DSMIT-specific (modeled after MIT's Kerberos) communication protocols. It provides an ongoing secure DSMIT operation, and supports secure modification of the security configuration and updates of passwords and keys.

DSMIT can also be used in an SP environment for doing system administration of the nodes from the CWS. In this section we discuss how to update the PTFs on the nodes from the CWS using DSMIT. DSMIT is a priced LPP.

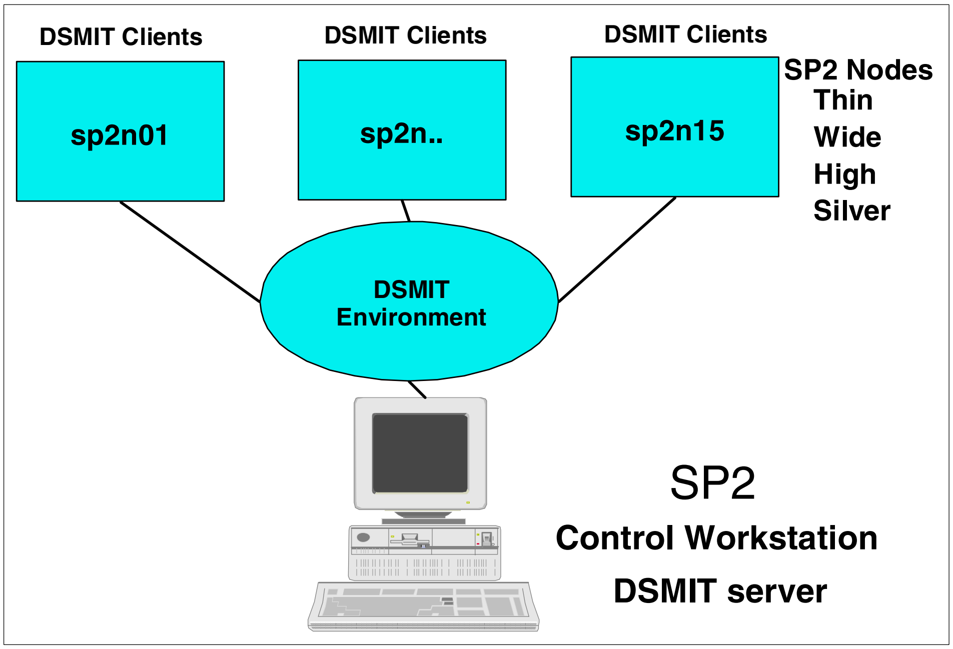


Figure 19. DSMIT Configuration in an SP Environment

DSMIT runs in both concurrent and sequential modes. Concurrent mode means that the DSMIT server builds a command and routes it to the clients simultaneously. Sequential mode means that the DSMIT server builds a command and routes it to the clients one machine at a time. After you build a command on the server and press the Enter key, a menu appears asking in which mode you wish to run DSMIT.

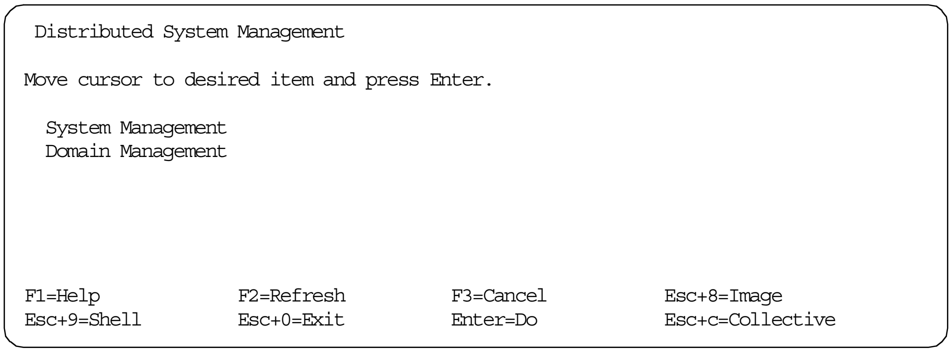
Using DSMIT, it is easier to do system administration than by using dsh from the CWS. The main reason for this is because, when using dsh, you have to know the options that are available for executing the commands, which you need not know when you are using DSMIT.

In the lab, we configured the CWS as the DSMIT server and all the nodes as DSMIT clients.

To start DSMIT with a working collection of two clients (sp2n10 and sp2n11), type:

```
# dsmitty -w sp2n10,sp2n11
```

If your credentials have expired or if you are logging on for the first time, it will prompt you for the administrator password.



```
 Distributed System Management

Move cursor to desired item and press Enter.

  System Management
  Domain Management






F1=Help             F2=Refresh          F3=Cancel           Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do            Esc+c=Collective
```

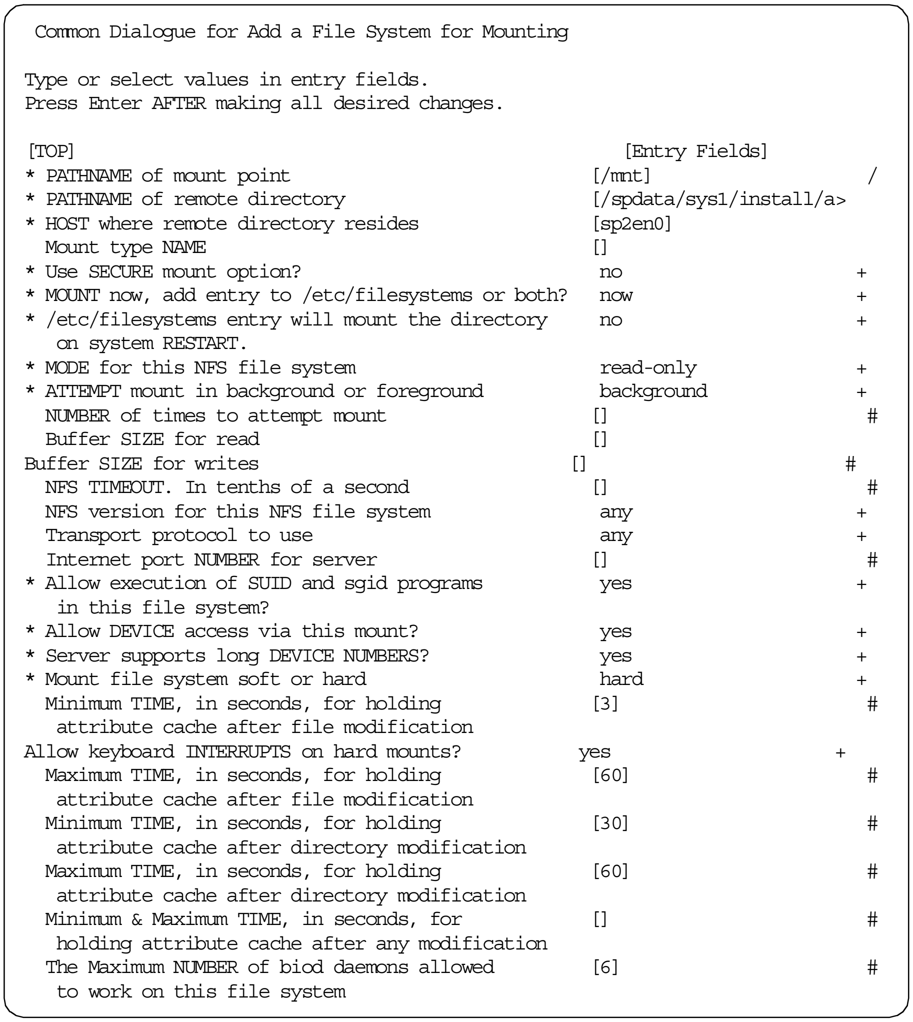
Select the option **System Management** to get into the SMIT screen, or **Domain Management** to get into DSMIT administration. Once you get into the system management, whatever operation you perform will be executed on clients sp2n10 and sp2n11. If you had selected more than one client, it will ask you whether you want to execute the command in sequence on all the clients, or if you want them to be executed concurrently.

Now let us discuss how to update the AIX PTFs to nodes sp2n10 and sp2n11 using DSMIT.

1. We have to mount the lppsource of the CWS to clients sp2n10 and sp2n11:

   ```
   # dsmitty -w sp2n10,sp2n11 mknfsmnt
   ```

   You will get a screen similar to the following:



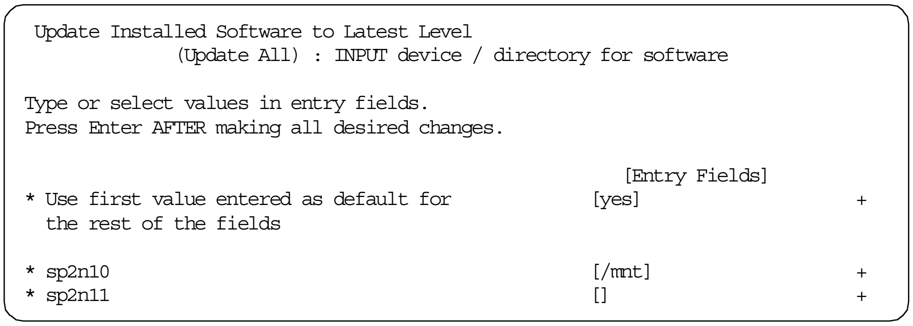
```
 Common Dialogue for Add a File System for Mounting

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
* PATHNAME of mount point                             [/mnt]                   /
* PATHNAME of remote directory                        [/spdata/sys1/install/a>
* HOST where remote directory resides                 [sp2en0]
  Mount type NAME                                     []
* Use SECURE mount option?                             no                     +
* MOUNT now, add entry to /etc/filesystems or both?    now                    +
* /etc/filesystems entry will mount the directory      no                     +
   on system RESTART.
* MODE for this NFS file system                        read-only              +
* ATTEMPT mount in background or foreground            background             +
  NUMBER of times to attempt mount                    []                       #
  Buffer SIZE for read                                []
Buffer SIZE for writes                              []                       #
  NFS TIMEOUT. In tenths of a second                  []                       #
  NFS version for this NFS file system                 any                    +
  Transport protocol to use                            any                    +
  Internet port NUMBER for server                     []                       #
* Allow execution of SUID and sgid programs            yes                    +
   in this file system?
* Allow DEVICE access via this mount?                  yes                    +
* Server supports long DEVICE NUMBERS?                 yes                    +
* Mount file system soft or hard                       hard                   +
  Minimum TIME, in seconds, for holding               [3]                      #
   attribute cache after file modification
Allow keyboard INTERRUPTS on hard mounts?            yes                    +
  Maximum TIME, in seconds, for holding               [60]                     #
   attribute cache after file modification
  Minimum TIME, in seconds, for holding               [30]                     #
   attribute cache after directory modification
  Maximum TIME, in seconds, for holding               [60]                     #
   attribute cache after directory modification
  Minimum & Maximum TIME, in seconds, for             []                       #
   holding attribute cache after any modification
  The Maximum NUMBER of biod daemons allowed          [6]                      #
   to work on this file system
```

When you press Enter, it will prompt for sequential or concurrent mode. Make your choice and press Enter. This mounts the sp2en0:/spdata/sys1/install/aix432/lppsource to /mnt in both clients sp2n10 and sp2n11. You can verify that the mount has happened by entering:

```
# dsh -w sp2n10,sp2n11 df /mnt
```

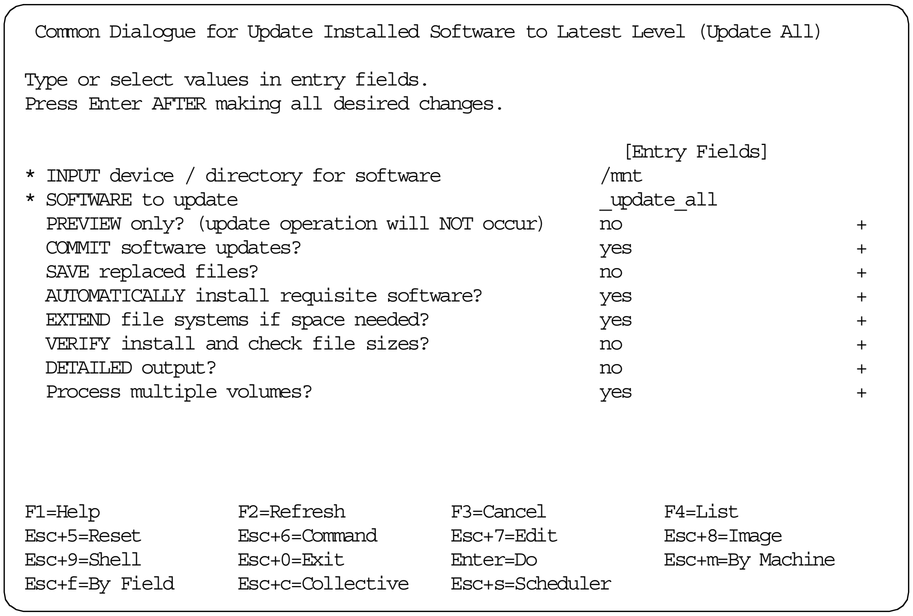
2. Install the PTFs in nodes sp2n10 and sp2n11:

```
# dsmitty -w sp2n10,sp2n11 update_all
```



```
 Update Installed Software to Latest Level
             (Update All) : INPUT device / directory for software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Use first value entered as default for             [yes]                  +
  the rest of the fields

* sp2n10                                             [/mnt]                 +
* sp2n11                                             []                     +
```

Input the Entry fields as shown in the preceding screen and press Enter to take you to the next screen.



```
 Common Dialogue for Update Installed Software to Latest Level (Update All)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* INPUT device / directory for software               /mnt
* SOFTWARE to update                                  _update_all
  PREVIEW only? (update operation will NOT occur)     no                    +
  COMMIT software updates?                            yes                   +
  SAVE replaced files?                                no                    +
  AUTOMATICALLY install requisite software?           yes                   +
  EXTEND file systems if space needed?                yes                   +
  VERIFY install and check file sizes?                no                    +
  DETAILED output?                                    no                    +
  Process multiple volumes?                           yes                   +




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do            Esc+m=By Machine
Esc+f=By Field      Esc+c=Collective    Esc+s=Scheduler
```

When you press Enter, it will prompt for sequential or concurrent mode. Make your choice and press Enter. This command is passed to clients sp2n10 and sp2n11 and the PTFs are installed in both systems. In this way, you can select all systems in one effort in order to install the PTFs in the nodes.

5.4 Applying PSSP PTFs in the CWS

The steps for applying the PSSP PTFs are as follows:

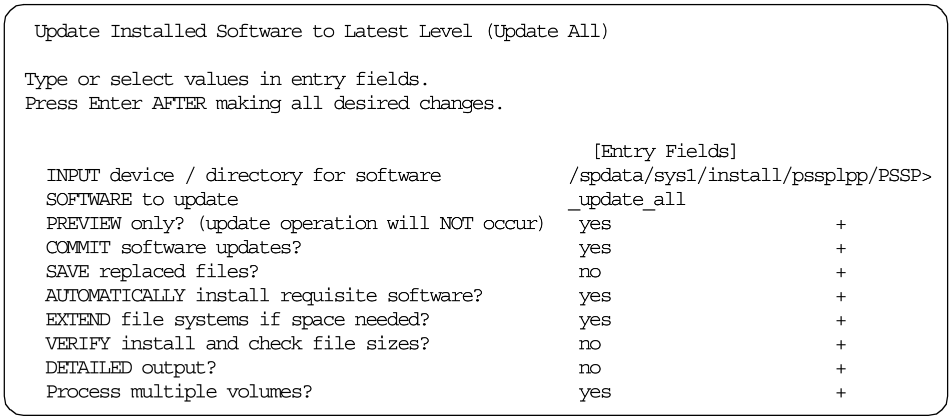
1. Create a mksysb backup image of the CWS. We recommend that you take two backups, on two different tapes, and never trust just one tape. Check that the tape is OK by listing the contents of the tape using the command `smitty lsmksysb`. For more information on taking a mksysb backup, refer to 7.1.1, “Mksysb and Savevg” on page 139.
2. Copy the PTFs to an appropriate directory. For example, in the lab we copied the files to the directory /spdata/sys1/install/pssplpp/PSSP-3.1.
3. Create a new .toc file by running the command:

   ```
   # inutoc /spdata/sys1/install/pssplpp/PSSP-3.1
   ```

4. Check the READ THIS FIRST that comes with any updates to the PSSP and .info files for the prerequisites, corequisites, and any precautions that need to be taken for installing these PTFs. Check the filesets in the directory you copied to verify that all the required filesets are available. If there are any prerequisite AIX PTFs, refer to 5.2, “Applying AIX PTFs in the CWS” on page 104 for installing AIX PTFs on the CWS.
5. Update the new PTFs to the CWS by using:

   ```
   # smitty update_all
   ```

   Input device for directory/software: `/spdata/sys1/install/pssplpp/PSSP-3.1`



```
 Update Installed Software to Latest Level (Update All)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                   [Entry Fields]
  INPUT device / directory for software          /spdata/sys1/install/pssplpp/PSSP>
  SOFTWARE to update                             _update_all
  PREVIEW only? (update operation will NOT occur)  yes                    +
  COMMIT software updates?                        yes                    +
  SAVE replaced files?                            no                     +
  AUTOMATICALLY install requisite software?       yes                    +
  EXTEND file systems if space needed?            yes                    +
  VERIFY install and check file sizes?            no                     +
  DETAILED output?                                no                     +
  Process multiple volumes?                       yes                    +
```

When the update is finished, go through the output messages for any specific problem. Based on these messages, you must take appropriate actions, depending on your environment.

6. Verify the correct operation of all SP and AIX control functions. For more information, refer to Chapter 3, “Verifying Your SP” on page 67.

5.5 Applying PSSP PTFs to the Nodes

Applying PSSP PTFs to the nodes can be done by using the same three different methods that we used in applying AIX PTFs. Before applying the PTFs, take a mksysb backup of the node. For more information on taking a mksysb backup, refer to 7.1.1, “Mksysb and Savevg” on page 139. The three different methods are:

1. Creating a new image that contains all the PSSP PTFs in one node and propagating the image to the nodes.
2. Mounting the lppsource directory from the CWS to the nodes and installing it manually by using `smitty update_all` or by using `dsh` commands from the CWS for all nodes.
3. If you have a `dsmit` package installed, then you can install using `dsmitty update_all` to all nodes in one effort.

5.5.1 Method 1: Applying PSSP PTFs Using mksysb Install Method

The procedure to install PSSP PTFs on the nodes using this method is the same as that used for installing AIX PTFs in 5.3.1, “Method 1: Applying AIX PTFs Using mksysb Install Method” on page 106.

For installing the PSSP PTFs, follow the same procedure except for Step 2: you will have to mount the PSSP PTF directory instead of the lppsource directory. The command is:

```
#mount sp2en0:/spdata/sys1/install/pssplpp/PSSP-3.1 /mnt
```

5.5.2 Method 2: Applying PSSP PTFs Using SMIT or dsh

This method is to be used for installing PSSP PTFs on the nodes using SMIT or `dsh` commands. The procedure to install the PSSP PTFs is the same as that for applying AIX PTFs in the nodes. Refer to 5.3.2, “Method 2: Applying AIX PTFs Using SMIT or dsh” on page 111.

Follow the same procedure except for Step 2: in both Option 1 and Option 2, you have to mount the PSSP PTF directory instead of the lppsource directory. The command is:

```
#mount sp2en0:/spdata/sys1/install/pssplpp/PSSP-3.1 /mnt
```

When updating the ssp.css fileset of PSSP, you must reboot the nodes in order for the kernel extensions to take effect.

5.5.3 Method 3: Applying PSSP PTFs Using DSMIT

This method is to be used for installing the PSSP PTFs on to the nodes using DSMIT, assuming DSMIT is installed in the CWS and the nodes. The procedure for installing is the same as that for installing AIX PTFs to the nodes. Refer to 5.3.3, “Method 3: Applying AIX PTFs Using DSMIT” on page 113.

Follow the same procedure except for Step 1: instead of mounting the lppsource directory, you will have to give the path of the PSSP ptfs directory. In the SMIT dialogue panel for Add a File System for Mounting, enter the pathname of the remote directory as /spdata/sys1/install/pssplpp/PSSP-3.1.

When updating the ssp.css fileset of PSSP, you must reboot the nodes in order for the kernel extensions to take effect.

5.6 Applying PTFs in the Nodes Using the Switch

In a larger SP configuration, installing PTFs using the Ethernet is very slow and time-consuming. In such cases, it is possible to install the PTFs by using the switch network, which is much faster compared to the Ethernet network. This method can be used for installing AIX PTFs. It is also possible to install the PSSP PTFs by using the Switch, since the installation process of the PTFs does not affect the drivers already loaded in the switch adapter. This method was used at some customer sites but nevertheless it is an unsupported method.

The steps for installing the PTFs by using the Switch are as follows:

1. Create a filesystem or directory in one of the nodes. Check that you have sufficient space to copy the files to this filesystem. In the lab, we created file system /ptf in node sp2n07.
2. NFS-export this filesystem to the CWS with read-write permission by using the command:

   ```
   # dsh -w sp2n07 /usr/sbin/mknfsexp -d '/ptf' -t 'rw' -r 'sp2en0' '-B'
   ```

3. Mount the exported filesystem of the node in the CWS on /mnt by using the command:

   ```
   # mount sp2n07:/ptf /mnt
   ```

4. Copy the PTFs from the tape to /mnt.

5. Initialize the table of contents .toc file by using the command:

```
# inutoc /mnt
```

6. Umount the filesystem from the CWS by using the command:

```
# umount /mnt
```

7. NFS-export the /ptf filesystem to all the SP nodes through the Switch network by using the command:

```
# dsh -w sp2n07 /usr/sbin/mknfsexp -d '/ptf' -t 'ro' -r
'sp2css01,sp2css05,sp2css06,sp2css07,sp2css08,sp2css09,sp2css10,sp2css1
1,sp2css12,sp2css13,sp2css14,sp2css15 ' '-B'
```

8. NFS-mount the /ptf file system of sp2n07 through the Switch on the node where the PTF is to be installed. For example, to install the PTFs on node sp2n08, execute the following command:

```
# dsh -w sp2n08
```

```
dsh> mount sp2css07:/ptf /mnt
```

9. Install the PTFs in node sp2n08 by using the command:

```
dsh>/usr/lib/instl/sm_inst installp_cmd -a -d '/mnt' -f '_update_all'
'-cNgX’
```

10. Unmount the filesystem that you mounted in Step 8 by using the command:

```
dsh> umount /mnt
```

Chapter 6. Migrating PSSP and AIX to Later Versions

This chapter covers the migration of PSSP and AIX to later versions in an SP environment. The objective is to help you to prepare and perform the migration of PSSP and AIX. Before migrating a production system, there are several things to be checked in order to perform a successful migration. In this chapter, we describe in detail the steps from the planning stage to the implementation of migration of PSSP software.

There are several things you need to do, or check, before starting the migration. The following list gives most of these items, but it is not exhaustive.

1. Get the relevant documentation, like READ THIS FIRST document for PSSP, and read it fully before performing the migration. The README files for all PSSP versions and their LPPs are available at the following URL: `http://www.rs6000.ibm.com/support/sp/sp_secure/readme/`
2. Check and document the AIX and PSSP levels of your CWS and all nodes.
3. Check your installed LPPs in the CWS and the nodes and document them.
4. Look for the required minimum AIX and PSSP PTF levels for the various AIX and PSSP versions.
5. Check the PTF levels for AIX and PSSP that are installed in the CWS and the nodes. Find out the PTFs that are needed to be installed for AIX, PSSP and other LPPs.
6. Understand the issues related to coexistence of various PSSP and AIX levels.
7. Choose the appropriate method (update/migrate/install) for upgrading your CWS and the nodes.
8. Check your disk space, and if necessary, get some extra disk space.
9. Document your system and the migration plan.
10. Estimate the migration time.
11. Prepare for recovery procedures, in case of unsuccessful migration.
12. Back up the rootvg of the CWS and the nodes. Also back up the files in the /spdata directory (directory backup, file system backup or savevg backup) of the CWS depending upon your configuration.

Note: For more detailed migration information, refer to *IBM Parallel System Support Programs for AIX: Installation and Migration Guide,* GC23-3898.

The basic flow of steps for doing a PSSP migration is shown in Figure 20 on page 124 and Figure 21 on page 125.

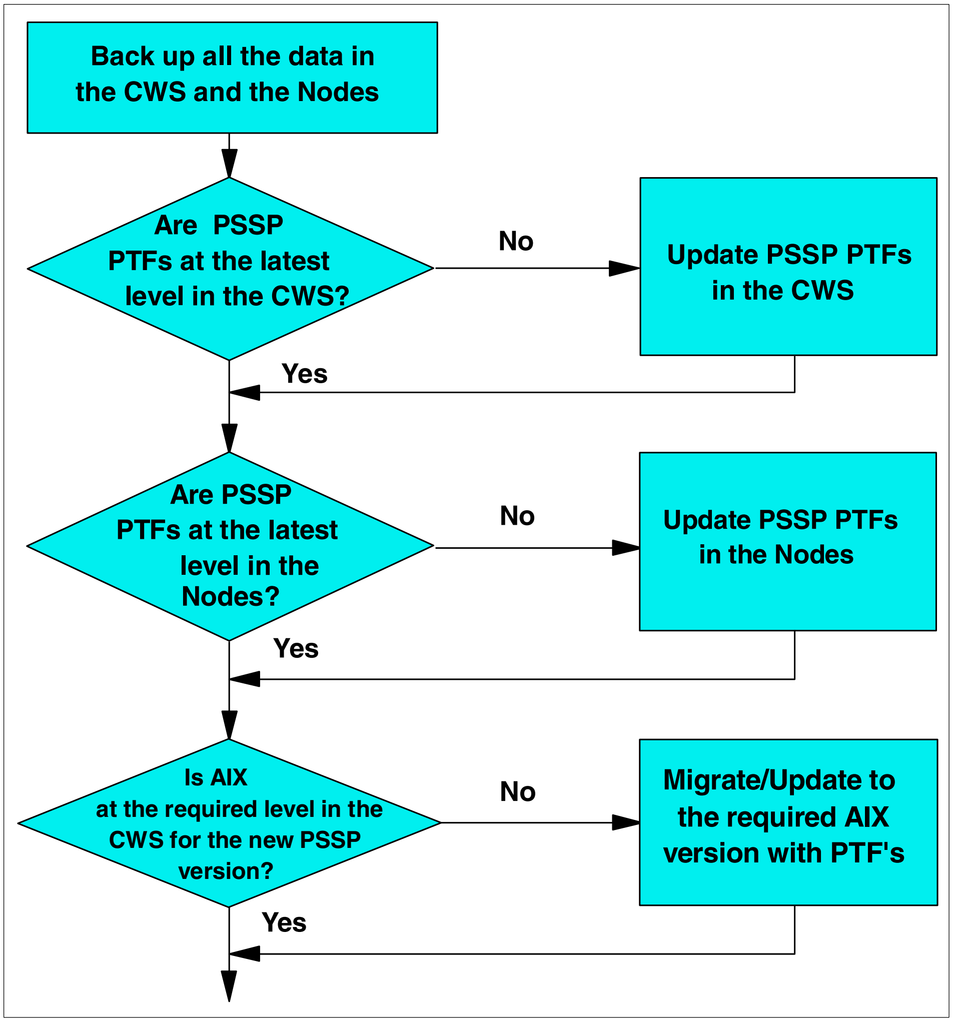

Back up all the data in the CWS and the Nodes
Are PSSP PTFs at the latest level in the CWS?
No
Update PSSP PTFs in the CWS
Yes
Are PSSP PTFs at the latest level in the Nodes?
No
Update PSSP PTFs in the Nodes
Yes
Is AIX at the required level in the CWS for the new PSSP version?
No
Migrate/Update to the required AIX version with PTF's
Yes

Figure 20. Basic Flowchart for PSSP Migration - Part 1

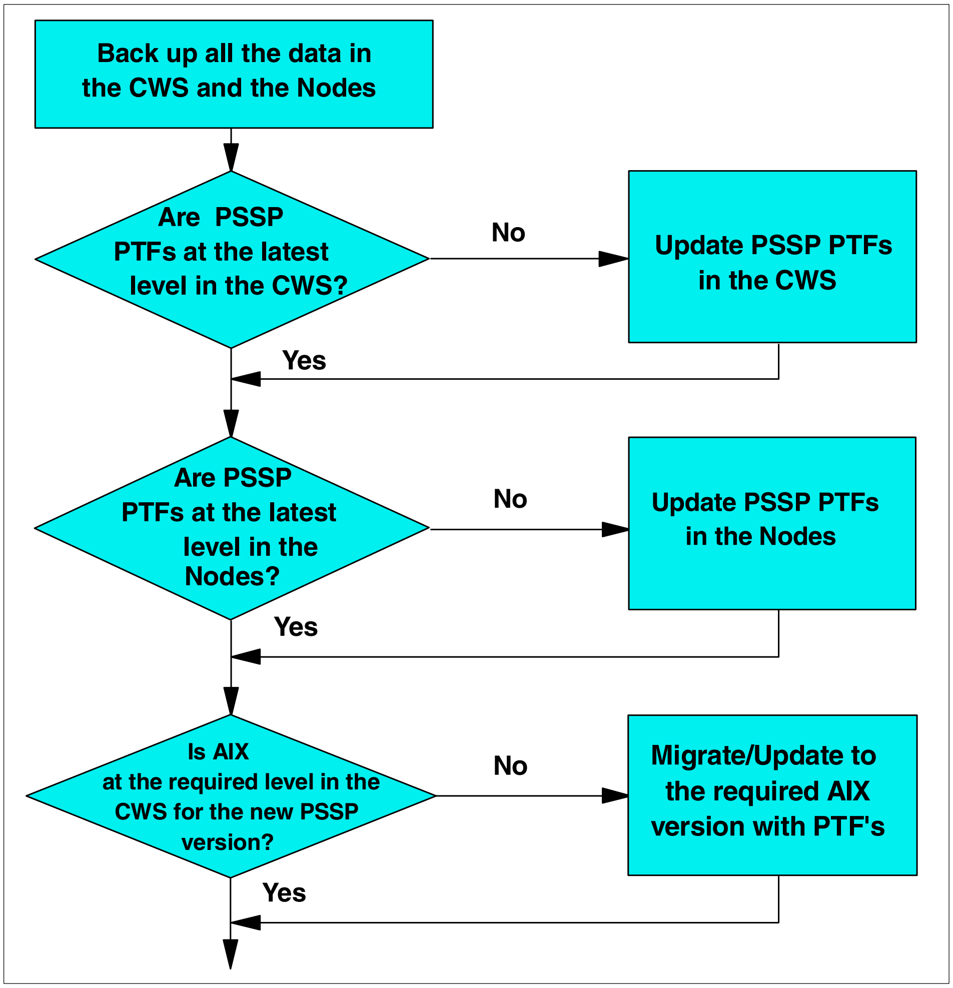

Back up all the data in the CWS and the Nodes
Are PSSP PTFs at the latest level in the CWS?
No
Update PSSP PTFs in the CWS
Yes
Are PSSP PTFs at the latest level in the Nodes?
No
Update PSSP PTFs in the Nodes
Yes
Is AIX at the required level in the CWS for the new PSSP version?
No
Migrate/Update to the required AIX version with PTF's
Yes

Figure 21. Basic Flowchart for PSSP Migration - Part 2

6.1 Coexistence

When you are planning to have different versions of PSSP software on the nodes, you must keep in mind that there are minimum PTF requisites for PSSP and AIX levels on the CWS and the nodes for it to work. This information is available in the READ THIS FIRST document that is provided with the PSSP software. For the README files for all the PSSP versions and their LPPs, refer to the following URL:

`http://www.rs6000.ibm.com/support/sp/sp_secure/readme/`

The minimum PTF requisites for coexistence/migration for various PSSP versions at the time of writing are shown in the Table 6. To get the latest update on the PTF levels for PSSP, refer to the following URL:

`http://www.rs6000.ibm.com/support/sp/sp_secure/status/servstat.html`

Table 6. PTF Requisites for Coexistence/Migration of PSSP and AIX Levels

Coexistence / Migration		
PSSP 2.4 on CWS	PSSP 1.2 Nodes	Not Supported
	PSSP 2.1 Nodes	PTF Set 28.
	PSSP 2.2 Nodes	PTF Set 15.
	PSSP 2.3 Nodes	PTF Set 6.
PSSP 2.3 on CWS	PSSP 1.2 Nodes	PTF Set 23.
	PSSP 2.1 Nodes	PTF Set 23 If 4.1.4, then add IX58039 + IX60723 + IX60742.
	PSSP 2.2 Nodes	PTF Set 11 + IX69336 + IX69491. If 4.1.4, then add IX58039 + IX60723 + IX60742. For RVSD 1.2 Nodes, IX69650.
PSSP 2.2 on CWS	PSSP 1.2 Nodes	PTF Set 23 + IX60961 or PTF Set 24 (which includes IX60961).
	PSSP 2.1 Nodes	PTF Set 18. If LoadLeveler 1.2.0, PTF Set 12. If LoadLeveler 1.2.1, PTF Set 5.

6.2 Migrating the CWS from PSSP 2.4 to PSSP 3.1

In the SP environment, the CWS should be at the same or at a higher level of AIX and PSSP than nodes. If there is already a higher level of AIX and/or PSSP available you should consider updating/migrating your CWS to this level. This will make future updates and/or migrations easier. So when you consider migrating your SP to a later version, the first step is to migrate the CWS to the required level of AIX and PSSP Versions. IBM suggests installing PTF service when upgrading AIX modification levels or when migrating to just a new PSSP level.

To migrate from PSSP 2.4 to PSSP 3.1 with the AIX version at 4.3.2, we suggest doing it as an upgrade service.

Note: Use this section in conjunction with *IBM Parallel System Support Programs for AIX: Installation and Migration Guide,* GC23-3898 when migrating the CWS.

The steps for migrating the CWS from PSSP 2.4 to PSSP 3.1 with AIX version at 4.3.2 are as follows:

1. Create a mksysb backup image of the CWS. We recommend that you always take two backups on two different tapes, and never trust just one tape. Check the tape is OK by listing the contents of the tape to be readable by using the command `smitty lsmksysb`. For more information on mksysb backup, refer to 7.1.1, “Mksysb and Savevg” on page 139.
2. Check that there is sufficient space in the /tftpboot and /spdata file systems. See 2.4.2, “Disk Space Considerations” on page 50 for more details.
3. Create directory /spdata/sys1/install/pssplpp/PSSP-3.1 by using:

   ```
   # mkdir /spdata/sys1/install/pssplpp/PSSP-3.1
   ```

4. Copy the PSSP 3.1 images into the /spdata/sys1/install/pssplpp/PSSP-3.1 directory. Rename the pssp package to pssp.installp and create the .toc file by using `the inutoc` command. The files related to PSSP-3.1 are:
   - pssp.installp
   - rsct.basic
   - rsct.clients
   - ssp.resctr
5. Stop the daemons in the CWS. Execute the following commands in the same sequence to stop the daemons in the CWS:

```
# syspar_ctrl -G -k

# stopsrc -s sysctld

# stopsrc -s splogd

# stopsrc -s hardmon

# stopsrc -g sdr
```

6. Check that all these daemons are stopped by using the command `lssrc -a`. They should be in inoperative state.
7. Update PSSP version 3.1 on the CWS by using the commands:

   ```
   # cd /spdata/sys1/install/pssplpp/PSSP-3.1

   # /usr/lib/instl/sm_inst installp_cmd -a -Q -d '.' -f '_all_latest'
   '-cNgXG’
   ```

   The list of ssp filesets installed are as follows:

   ```
   # lslpp -L | grep -E "ssp|rsct"
     rsct.basic.hacmp           1.1.0.2    A    RS/6000 Cluster Technology basic
     rsct.basic.rte             1.1.0.4    A    RS/6000 Cluster Technology basic
     rsct.basic.sp              1.1.0.3    A    RS/6000 Cluster Technology basic
     rsct.clients.hacmp         1.1.0.0    C    (ECIP) RS/6000 Cluster
     rsct.clients.rte           1.1.0.2    A    RS/6000 Cluster Technology
     rsct.clients.sp            1.1.0.0    C    (ECIP) RS/6000 Cluster
     ssp.authent                3.1.0.1    A    SP Authentication Server
     ssp.basic                  3.1.0.4    C    SP System Support Package
     ssp.clients                3.1.0.4    C    SP Authenticated Client Commands
     ssp.css                    3.1.0.4    C    SP Communication Subsystem
     ssp.docs                   3.1.0.0    C    SP man pages and PDF files and
     ssp.gui                    3.1.0.4    C    SP System Monitor Graphical User
     ssp.ha_topsvcs.compat      3.1.0.0    C    Compatability for ssp.ha and
                                                ssp.topsvcs clients
     ssp.jm                     3.1.0.1    C    SP Job Manager Package
     ssp.perlpkg                3.1.0.0    C    SP PERL Distribution Package
     ssp.pman                   3.1.0.1    A    SP Problem Management
     ssp.ptpegui                3.1.0.1    C    SP Performance Monitor Graphical
     ssp.public                 3.1.0.0    C    Public Code Compressed Tarfiles
     ssp.resctr.rte             3.1.0.0    C    SP Resource Center
     ssp.spmgr                  3.1.0.2    A    SP Extension Node SNMP Manager
     ssp.st                     3.1.0.1    A    Job Switch Resource Table
     ssp.sysctl                 3.1.0.1    A    SP Sysctl Package
     ssp.sysman                 3.1.0.2    A    Optional System Management
     ssp.tecad                  3.1.0.0    C    SP HA TEC Event Adapter Package
     ssp.top                    3.1.0.1    A    SP Communication Subsystem
     ssp.top.gui                3.1.0.1    C    SP System Partitioning Aid
     ssp.ucode                  3.1.0.1    A    SP Supervisor Microcode Package
     ssp.vsdgui                 3.1.0.1    C    VSD Graphical User Interface
   ```

8. Authenticate yourself as the administrative user to the authentication database by using the command:

```
# kinit root.admin
```

9. Complete the PSSP installation on the CWS by using the command as shown in the following screen:

```
# install_cw
0 objects deleted
0513-044 The stop of the hardmon Subsystem was completed successfully.
0513-083 Subsystem has been Deleted.
0513-071 The hardmon Subsystem has been added.
An entry for /usr/lpp/ssp/bin/sdrd already exists in inittab.
An entry for hardmon already exists in inittab.
0513-004 The Subsystem or Group, splogd, is currently inoperative.
0513-083 Subsystem has been Deleted.
0513-071 The splogd Subsystem has been added.
0513-059 The splogd Subsystem has been started. Subsystem PID is 18156.
Stopping the spmgr subsystem is currently running
An entry for the spmgr subsystem already exists in inittab.
Starting the spmgr subsystem
Starting the swtlog subsystem
Starting the swtadmd subsystem
0513-059 The sysctld Subsystem has been started. Subsystem PID is 26764.
install_cw: If you want to bring up the Perspectives Launch Pad to help
 you complete your installation, enter "perspectives &" now.
#
```

10. Verify the SDR and system monitor for correct installation by using the following commands:

```
# SDR_test
SDR_test: Start SDR commandline verification test
SDR_test: Verification succeeded
# spmon_itest
spmon_itest: Start spmon installation verification test
spmon_itest: Verification Succeeded
#
```

11. Set up the site environment in the CWS for the AIX level by using the command:

```
# spsitenv cw_lppsource_name=aix432
```

12. Start the system management environments on the CWS by using command services_config; the output should look looks as follows:

```
# /usr/lpp/ssp/install/bin/services_config
rc.ntp: NTP already running - not starting ntp
0513-029 The supfilesrv Subsystem is already active.
Multiple instances are not supported.
/etc/auto/startauto: The automount daemon is already running on this system.
#
```

13.Check whether the supervisor microcode needs to be updated by using the following command:

```
# spsvrmgr -G -r status all

spsvrmgr: Frame  Slot  Supervisor  Media         Installed     Required
                       State       Versions      Version       Action
          _____  ____  __________  ____________  ____________  ____________
          1      0     Active      u_10.1c.0709  u_10.1c.070c  None
                                   u_10.1c.070c

                 ____  __________  ____________  ____________  ____________
                 1     Active      u_10.3a.0614  u_10.3a.0614  Upgrade
                                   u_10.3a.0615

                 ____  __________  ____________  ____________  ____________
                 17    Active      u_80.19.060b  u_80.19.060b  None
#
```

14.The output of the previous command indicates that microcode for high node in frame-1, slot-1 needs to be updated. Update the microcode by using the following command:

```
0)sp2en0:/ 89$ spsvrmgr -G -u 1:1

spsvrmgr: Dispatched "microcode" process [28506] for frame 1 slot 1.
          Process will take approximately 5 minutes to complete.
spsvrmgr: Process [28506] for frame 1 slot 1 completed successfully.
```

15.In PSSP-3.1, the authenticated r-commands in the AIX 4.3.2 operating system are used. Older PSSP versions installed their own r-command (in the subdirectory /usr/lpp/ssp/rcmd/bin). Therefore, you have to select an authentication method and activate it afterwards.

1. Select Authentication Method.

   If you are using SMIT, issue the following command:

   ```
   # smitty spauth_config
   ```

   Select **Select Authorization Methods.**

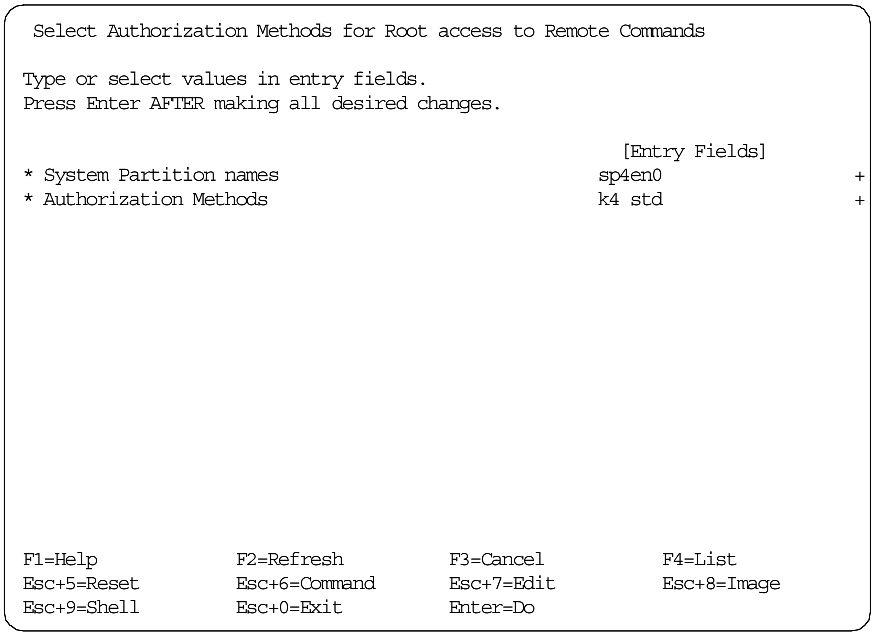
   Select **System Partition name.**

Move the cursor to the desired partition.

Select **Authorization Methods.**

Select one or more methods by using PF7.

See the following screen for an example.



```
 Select Authorization Methods for Root access to Remote Commands

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* System Partition names                              sp4en0            +
* Authorization Methods                               k4 std            +

















F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

You can also select the appropriate authentication methods by issuing the following command:

```
# spsetauth -d -p sp4en0 k4 std
```

2. Enable Authentication Method

   The next step is to enable the authentication method. If you are using SMIT, issue the following command:

   ```
   # smitty spauth_config
   ```

   Select **Enable Authorization Methods.**

   Select **Enable on the Control Workstation only.**

   Select **Force change on nodes.**

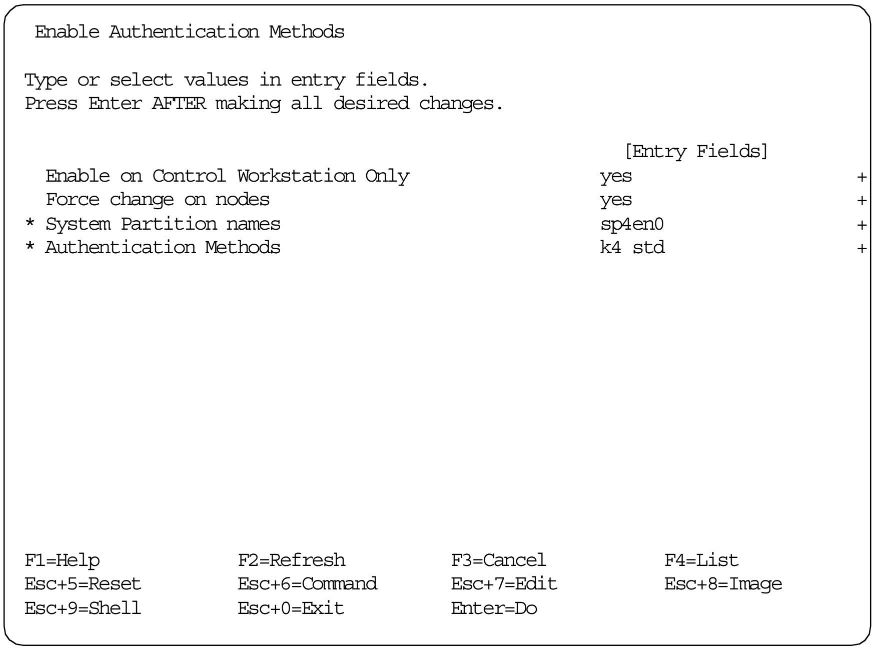
   Select the option **System Partition name.**

   Move the cursor to the desired partition.

   Select **Authentication Methods.**

Select one or more methods by using PF7.

See the following screen for an example.



```
 Enable Authentication Methods

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Enable on Control Workstation Only                  yes                 +
  Force change on nodes                               yes                 +
* System Partition names                              sp4en0              +
* Authentication Methods                              k4 std              +











F1=Help            F2=Refresh         F3=Cancel          F4=List
Esc+5=Reset        Esc+6=Command      Esc+7=Edit         Esc+8=Image
Esc+9=Shell        Esc+0=Exit         Enter=Do
```

You can also enable the appropriate authentication methods by issuing the following command:

```
# chauthpar [-c] [-f] -p sp4en0 k4 std
```

16. There are system partition-sensitive subsystems that need to be added to the system. Even if you are not partitioning your system, you need to do this since you still have one default partition. Do the following:

    1. Remove the old subsystems by using the following command:

       ```
       # syspar_ctrl -c
       ```

    2. Add new subsystems by issuing the following command:

       ```
       # syspar_ctrl -A -G
       ```

    The following screen shows an example command output:

```
# syspar_ctrl -c
hr.sp2en0 susbsystem has been deleted

(0)sp2en0:/ 92$ syspar_ctrl -A -G
0513-071 The hats.sp2en0 Subsystem has been added.
0513-071 The hags.sp2en0 Subsystem has been added.
0513-071 The hagsglsm.sp2en0 Subsystem has been added.
0513-071 The haem.sp2en0 Subsystem has been added.
0513-071 The haemaixos.sp2en0 Subsystem has been added.
Deleted 370 objects from class EM_Resource_Variable
Added 370 objects to class EM_Resource_Variable
Added 45 objects to class EM_Resource_ID
Added 54 objects to class EM_Resource_Variable
Added 8 objects to class EM_Structured_Byte_String
Added 3 objects to class EM_Resource_ID
Added 2 objects to class EM_Resource_Class
Added 2 objects to class EM_Resource_Monitor
haemcfg: Reading Event Management data for partition: sp2en0.
haemcfg: Created EMCDB file: /spdata/sys1/ha/cfg/em.sp2en0.cdb Version: 90070676
3,274389248,0.
making SRC object "hr.sp2en0"
0513-071 The hr.sp2en0 Subsystem has been added.
0513-071 The pman.sp2en0 Subsystem has been added.
0513-071 The pmanrm.sp2en0 Subsystem has been added.
0513-071 The Emonitor.sp2en0 Subsystem has been added.
0513-059 The hats.sp2en0 Subsystem has been started. Subsystem PID is 15164.
0513-059 The hags.sp2en0 Subsystem has been started. Subsystem PID is 20478.
0513-059 The hagsglsm.sp2en0 Subsystem has been started. Subsystem PID is 19664.
0513-059 The haem.sp2en0 Subsystem has been started. Subsystem PID is 33212.
0513-059 The haemaixos.sp2en0 Subsystem has been started. Subsystem PID is 23214
.
0513-059 The hr.sp2en0 Subsystem has been started. Subsystem PID is 28522.
0513-059 The pman.sp2en0 Subsystem has been started. Subsystem PID is 17828.
0513-059 The pmanrm.sp2en0 Subsystem has been started. Subsystem PID is 17292.
0513-059 The sp_configd Subsystem has been started. Subsystem PID is 14680.
#
```

17. Verify that all the system partition subsystems have been properly started by using the following command:

```
# lssrc -a | grep sp2en0
 sdr.sp2en0       sdr              11354   active
 hats.sp2en0      hats             20386   active
 hags.sp2en0      hags             21160   active
 hagsglsm.sp2en0  hags             21680   active
 haem.sp2en0      haem             16004   active
 haemaixos.sp2en0 haem             17804   active
 hr.sp2en0        hr               15488   active
 pman.sp2en0      pman             20136   active
 pmanrm.sp2en0    pman             19880   active
 Emonitor.sp2en0  emon                     inoperative
#
```

18. The pmand daemons need to be refreshed in order to recognize the changes that were made to the SDR. To refresh, use the command:

    ```
    # dsh -avG refresh -s pman
    ```

    The output of the `refresh` command should look as follows for all the nodes defined in the SDR:

    ```
    sp2n01: pmand in partition sp2en0 refreshed, waiting events at Fri Jul
    17 16:22:02 1998

    sp2n01: 0513-095 The request for subsystem refresh was completed
    successfully.
    ```

19. Start the switch by giving the command `Estart` and verify whether all the nodes have joined the Switch.

20. Run all the verification tests that are appropriate to your SP environment for correct operation.

6.3 Migrating the Nodes from PSSP 2.4 to PSSP 3.1

In this example, we had an SP configuration with AIX version at 4.3.2 on the CWS and on the nodes, and a PSSP version at 2.4. The steps for migrating node sp2n06 from version PSSP 2.4 to PSSP 3.1 are as follows:

1. Create a mksysb backup image of the node. For more information on taking a mksysb backup, refer to 7.1.1, “Mksysb and Savevg” on page 139.

2. Enter the node configuration data to set the SDR for the new PSSP level to be installed by using the following two commands:

    ```
    # spchvgobj -r rootvg -p PSSP-3.1 -l 6
    spchvgobj: Successfully changed Node and Volume_Group objects for node number 6,
    volume group rootvg.
    spchvgobj: The total number of changes successfully completed is 1.
    spchvgobj: The total number of changes which were not successfully completed is 0.
    # spbootins -s no -r customize -l 6
    ```

3. Verify the values in the SDR for the correct pssp_ver by using the following command:

```
 splstdata -b -l 6
                 List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response           install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
-------------------------------------------------------------------------------
    6 sp2n06             10005AFA1B12    0    customize                 hdisk0
         bos.obj.ssp.432 Tue_Apr_13_17:29:51     bos.obj.ssp.432         aix432
                PSSP-3.1             rootvg
```

4. Now you can either run `setup_server` to initiate an install operation, or you can do it directly by using the wrapper-commands. Execute the following commands to prepare for the installation of node sp2n06.

   ```
   # setup_server
   ```

   or

   ```
   # Efence -G -autojoin sp2n06
   # create_krb_files
   # mkconfig
   # mkinstall
   # export_clients
   # allnimres -l 6
   ```

   After you execute these commands, the screen output should look as follows:

```
# Efence -G -autojoin sp2n06
All nodes successfully fenced.

# create_krb_files
create_krb_files: tftpaccess.ctl file and client srvtab files created/updated
on server node 0.

# mkconfig

# mkinstall

# export_clients
export_clients: File systems exported to clients from server node 0.

# llnimres -l 6
allnimres: Node 6 (sp2n06) prepared for operation: customize.
```

5. Refresh the system partition-sensitive subsystems by using the following command:

```
# syspar_ctrl -r -G
0513-095 The request for subsystem refresh was completed successfully.
0513-095 The request for subsystem refresh was completed successfully.
```

6. Copy the pssp script from the CWS to the /tmp directory of sp2n06 by using the command:

   `# pcp -w sp2n06 /spdata/sys1/install/pssp/pssp_script /tmp/pssp_script`

7. Execute the pssp script on node sp2n06 from the CWS by using the command:

   `# dsh -w sp2n06 /tmp/pssp_script`

   The output of this command should look as follows:

```
# dsh -w sp2n06 /tmp/pssp_script
sp2n06:
sp2n06:               =====================================================
sp2n06: pssp_script: = Making PSSP log directories...                     =
sp2n06:               =====================================================
sp2n06:
sp2n06:               =====================================================
sp2n06: pssp_script: = Switching output to log file...                    =
sp2n06:               =====================================================
sp2n06: + /usr/bin/mkdir -p /var/adm/SPlogs/sysman
sp2n06: + 1> /dev/null 2>& 1
sp2n06: + exec
sp2n06: + 3> /var/adm/SPlogs/sysman/NODE.config.log.7792
```

8. Check the log file /var/adm/SPlogs/sysman/sp2n06.config.log.7792 for any errors. Check that no lpp is in a broken state by using the command `lppchk -v`.

9. Check the bootp_response for this node and ensure that it is set to disk by using the following command:

```
1)sp2en0:/ 105$ splstdata -b -l 6
                  List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response           install_disk
      last_install_image   last_install_time  next_install_image lppsource_name
                pssp_ver         selected_vg
-------------------------------------------------------------------------------
    6 sp2n06            10005AFA1B12    0        disk                 hdisk0
         bos.obj.ssp.432 Tue_Apr_13_15:04:18     bos.obj.ssp.432       aix432
                PSSP-3.1              rootvg
```

10. Reboot the node to enable the kernel changes related to the new PSSP version to take effect.

11. Run all the verification tests that are appropriate to your SP environment for their correct operation.

Chapter 7. Backup and Recovery

As we know, there is no such thing as a “perfect” system. Any component of your system, independent of the quality of the product or of the products and mechanisms implemented to increase availability, is susceptible to failure. The data stored in your system could get lost due to human error or component failure. Hence, should such failures occur, you must have procedures and medias in place to reconstruct your environment as it was before those problems arose.

The whole business of a company or success of a project depends on successful backups, meaning the ability to recover all the data that has been stored in a machine, in a minimum time and as near as possible to the exact data that was there before the failure event.

Although the most important data to protect is the information of the business. it is also important to back up the configuration of your system so that you could continue working or restart your production environment in case of component failure.

This chapter helps you to understand how to back up and restore your whole SP2 system, as well as its subsystems: the SDR, the Kerberos database, normal directories and logical volumes.

7.1 Backing Up Your SP System

This section covers the procedures to back up the SP system and its various subsystems. To restore the SP system and its subsystems, refer to 7.2, “Restoring Your SP System” on page 147.

7.1.1 Mksysb and Savevg

The `mksysb` and `savevg` commands should be used to protect the whole system; through these tools you can save all the system data and user data.

The file-system image is in backup-file format. The tape format includes a boot image, a bosinstall image, and an empty table of contents, followed by the system backup (root volume group) image. The root volume group image is in backup-file format, starting with the data files and then any optional map files.

Keep the following things in mind when taking a backup with mksysb:

- The mksysb tool cannot back up raw devices, unmounted file systems, nfs filesystems, or volume groups differents to the rootvg. Make sure that you use a different tool to back up that data.
- Some rspc systems do not support booting from tape. When you make a bootable mksysb image on an rspc system that does not support booting from tape, the mksysb command issues a warning indicating that the tape will not be bootable.
- You can install a mksysb image from a system that does not support booting from tape by booting from a CD and entering maintenance mode; while in maintenance mode, you will be able to install the system backup from tape.

For a complete explanation of the `mksysb` command, refer to *IBM Parallel System Support Programs for AIX: Command and Technical Reference, Volume 1 and Volume 2,* SA22-7351*.*

7.1.1.1 Mksysb of the CWS

The CWS will normally have a tape device attached directly, so there is no difference in the way that you take a backup from a normal workstation.

You can generate the mksysb either through SMIT or by using the command line.

If you use SMIT, type the following command:

```
# smitty mksysb
```

If you use the command from command line and you want to generate the ./image.data file before taking the mksysb, type the following command:

```
# mksysb -i <device name>
```

If you do not want to generate the ./image.data file because you made a change to the existing one, you can omit the -i flag.

It is always important to have a recent backup of the CWS. You can decide to take one image periodically, for example each week or month, depending upon your environment, and take another copy each time you change the configuration.

Remember that `mksysb` will back up only the data contained in the rootvg volume group and that you will have to use savevg to back up the other volume groups.

7.1.1.2 Mksysb of a Node

Normally you do not have a tape drive attached to all your nodes, so therefore you are not able to take backups directly to tape. It will be necessary to use the network to transfer your backups to another machine with a connected tape drive attached.

In order to take an image of your node, follow the same steps mentioned in 7.1.1.1, “Mksysb of the CWS” on page 140. However, your image is usually created on your local node, but for later use you need to have it on your CWS; the preferred subdirectory is /spdata/sys1/install/images. As always, there is more than one solution to this problem.

- Create the image on the local node and transfer it afterwards to the CWS. You can use the `ftp` or `rcp` command.
- Export /spdata/sys1/install/images and mount it on your node. Create the image on this mounted filesystem.
- Create a named pipe on your system and use this pipe as the output device for the mksysb command. Start a remote copy; use the named pipe as the source and the file name of the image (in the subdirectory /spdata/sys1/install/images on the CWS) as the destination.

The image file is not bootable, so you will have to restore this as explained in 7.2.1.2, “Restoring a mksysb of a Node” on page 148.

In common system environments, you have some nodes with the same configuration working the same application. You can take a backup of one node of each group so that you can restore that image into any node of the same kind and purpose. For example, if you have 10 nodes working with AIX 4.3.2 and configured to work with Oracle, you could take a mksysb of one node. Then, if you need to restore the information of any of those nodes in the future, you can use the same mksysb to restore both AIX and also Oracle, and restore the Oracle configuration files of a specific node.

7.1.1.3 How to Verify a Mksysb

The only way to verify that there is absolutely no problem with a mksysb is to restore it in another machine. But if you cannot do that, you can verify the following: that the tape media can be read, that the bootable part of the tape is OK, and that the mksysb stored the files that you wanted. The following section describes how to do these verifications.

Data Verification

You can verify that you can read the data in the tape and store a copy of its contents, if desired, in the following way:

If you use SMIT, type:

```
# smitty lsmksysb
```

Using the command line:

```
# tctl -f /dev/rmt0 rewind
# restore -s4 -Tvqf /dev/rmt0.1 > /tmp/mksysb.lst
```

Boot Verification

The only way to verify that a mksysb tape will successfully boot is to bring the machine down and boot from the tape. No data needs to be restored.

Attention

Having the PROMPT field in the bosint.data file set to no causes the system to begin the mksysb restore automatically, using preset values with no user intervention.

If the state of PROMPT is unknown, this can be set during the boot process. After answering the prompt to select a console during the boot up, a rotating character is seen in the lower left of the screen. As soon as this character appears, type `000` and press Enter. This will set the prompt variable to yes.

You can also check the state of this variable while in normal mode by typing the following commands:

```
# chdev -l rmt0 -a block_size=512
# tctl -f /dev/rmt0 rewind
# cd /tmp
# restore -s2 -xvqf /dev/rmt0.1 ./bosinst.data
```

Then you can edit the file bosinst.data and check the PROMPT variable to find out its value.

7.1.1.4 Savevg

The `savevg` command finds and backs up all files belonging to a specified volume group. The volume group must be varied on, and the file systems must be mounted.

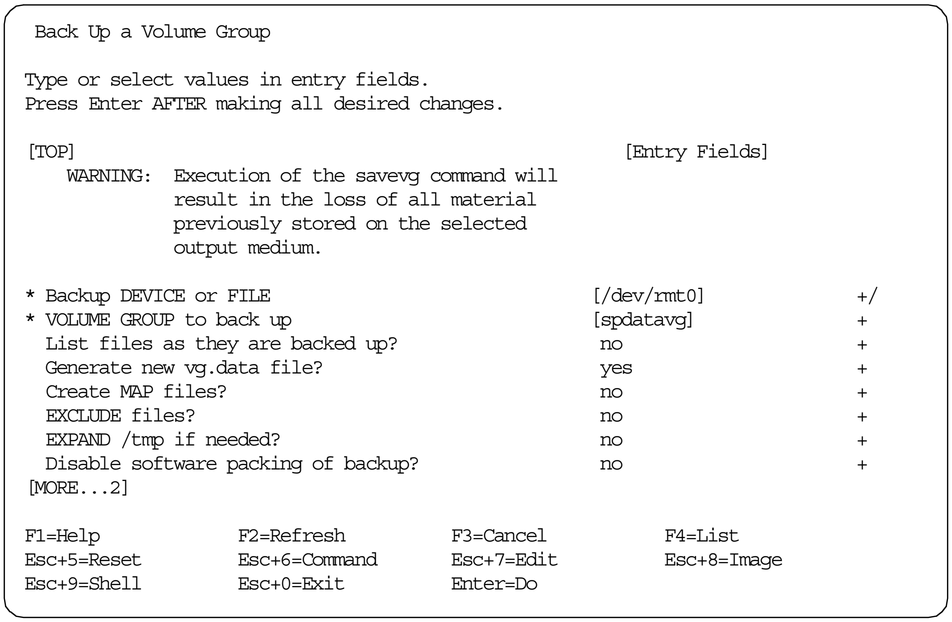
It is important to have a backup of the spdata file system. If you did not create an independent volume group to store this file system, and it is mounted in the rootvg file system, the spdata file system will be included in your CWS image created with the `mksysb` command. However, if you have the file system spdata in a volume group different from rootvg, you will have to use the savevg command to back up the volume group, or use some other tool to back up file systems independently.

There are two ways of using savevg, one through SMIT and the other through the command line.

To back up the splstdatavg using SMIT, type:

```
# smitty savevg
```

Figure 22 shows the result.



```
 Back Up a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
    WARNING:  Execution of the savevg command will
              result in the loss of all material
              previously stored on the selected
              output medium.

* Backup DEVICE or FILE                              [/dev/rmt0]            +/
* VOLUME GROUP to back up                            [spdatavg]             +
  List files as they are backed up?                   no                    +
  Generate new vg.data file?                          yes                   +
  Create MAP files?                                   no                    +
  EXCLUDE files?                                      no                    +
  EXPAND /tmp if needed?                              no                    +
  Disable software packing of backup?                 no                    +
[MORE...2]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Figure 22. Display of SMIT savevg Settings

If you are taking the backup in a node and you do not have a tape drive installed locally, you can choose one of the three methods described at the beginning of this section.

7.1.2 Backing Up the SDR

The SDR is required in order for your SP system to function, so you should have at least one backup of this database. The following task and subsystems depend on the SDR:

- The Job Manager

- VSD configuration tasks
- SP system configuration, customization and installation tasks
- The hardware monitor clients
- The SP nodes, when they are booted
- The SP Switch, when the switch is started
- The High Availability subsystem

You can take a backup of the SDR through SMIT or by using the command line, as follows.

If you use SMIT, type:

```
# smitty syspar
```

Select **Archive System Data Repository.**

Type the extension that you want for the filename. (This is optional). Press Enter.

Using the command line:

```
# SDRArchive <append_string>
```

One file will be generated in the directory /spdata/sys1/sdr/archives with the following format:

backup.JULIANdate.HHMM.append_string

JULIANdate.HHMM is a number or string uniquely identifying the date and time of the archive, and append_string is the argument entered in the command invocation, if specified.

The `SDRArchive` command is a script contained in the directory /usr/lpp/ssp/bin that executes a tar similar to the following:

```
# tar -cf /spdata/sys1/sdr/archives/backup.* /spdata/sys1/sdr/defs \
/spdata/sys1/sdr/sys
```

If you want, you can take a look to the file using the command:

```
# tar -tvf <file name>
```

To restore a savevg backup, use the `restvg` command as explained in 7.2.1.3, “The Restore Volume Group Command” on page 148.

7.1.3 Kerberos Database

The kerberos database is needed in order for the SP to work properly. If something goes wrong with kerberos, you can experience problems with the Switch, spmon, parallel commands and so forth. Therefore, you must have an up-to-date backup of your kerberos database.

Kerberos provides some commands that help you to take backups and do recovery of the kerberos database. The command used to backup is `kdb_util dump`, followed by the file name where you want to store the backup.

For example, if you want to back up the kerberos database to a file called kerberos.940422, you should execute:

```
# kdb_util dump kerberos.990422
```

The command will create a ASCII file called 990422 that has a format similar to the one shown in Figure 23 on 145. Although you can put this file anywhere, we recommend that you put it in the kerberos database directory called /var/kerberos/database.

```
hardmon sp4en0 255 1 1 0 fa50af36 5f82b6b6 203801010459 199902210245 root admin
rcmd sp4sw06 255 1 1 0 380cf1de f60552b8 203801010459 199902221651 root admin
K M 255 1 1 0 d0f57e3f 6d27fc15 203801010459 199902210242 db_creation *
rcmd sp4sw25 255 1 1 0 97ed8fe3 5bb4c135 203801010459 199904091434 root admin
rcmd sp4sw27 255 1 1 0 34b2c9ce d2be40c2 203801010459 199904091434 root admin
rcmd sp4n11 255 1 1 0 c71bc192 ce5c42e8 203801010459 199902221651 root admin
rcmd sp4n13 255 1 1 0 d6cc1011 853697a3 203801010459 199902221651 root admin
rcmd sp4n01 255 1 1 0 7c8addfe 643ec26c 203801010459 199902221651 root admin
rcmd sp4sw13 255 1 1 0 3db9c61e 2c3d995 203801010459 199902221651 root admin
rcmd sp4en1 255 1 1 0 dc07675e a010a874 203801010459 199902261624 root admin
krbtgt MSC.ITSO.IBM.COM 255 1 1 0 248e516a 53e2ad30 203801010459 199902210242 db
_creation *
rcmd sp4en0 255 1 1 0 3fd5361 69e766e9 203801010459 199902210245 root admin
q q 255 1 1 0 5910b883 9c531f43 203801010459 199903291519 * *
default * 255 1 1 0 0 0 203801010459 199902210242 db_creation *
rcmd sp4n31 255 1 1 0 2624cf4e 1237938b 203801010459 199904091434 root admin
rcmd sp4sw31 255 1 1 0 aef5ca0d fbd0d81d 203801010459 199904091434 root admin
rcmd sp4sw07 255 1 1 0 2d624a68 d4ec615 203801010459 199902221651 root admin
rcmd sp4sw21 255 1 1 0 9e81aa86 69b0acd1 203801010459 199904091434 root admin
rcmd sp4sw26 255 1 1 0 655e9862 9da6ec98 203801010459 199904091434 root admin
hardmon sp4cw0 255 1 1 0 53609c0b 71598274 203801010459 199902210245 root admin
...
```

Figure 23. Typical Format of a Kerberos Backup File

When you have secondary authentication servers defined and you follow the steps outlined for setting them up, you will automatically create a backup of the database every time the cron file entry runs that propagates the database to the secondary servers. You can verify if you have that entry in your crontab file by typing:

```
# crontab -l
```

The entry that propagates the database to the secondary servers and takes a backup of the database looks like the following:

```
0 * * * * /usr/kerberos/etc/push-kprop
```

If you have this entry, the push-kprop script will create a backup file called `slavesave` in the database directory /var/kerberos/database.

If you do not have this entry, but you want to automatically and periodically take backups of your kerberos database, you can include the `kdb_util` command to your crontab file.

7.1.4 Backing Up the NIM Database

The NIM database can be backed up from SMIT or by using the command line, as follows:

If you use SMIT, type:

```
# smitty nim_backup_db
```

Then enter the name of a device or a file to which the NIM database and the /etc/niminfo file will be backed up.

Using the command line:

Save the following NIM files using any method, for example, tar:

/etc/niminfo

/etc/objrepos/nim_attr

/etc/objrepos/nim_attr.vc

/etc/objrepos/nim_object

/etc/objrepos/nim_object.vc

When you use SMIT, you can see that a tar is executed to back up these files. Do not forget that a backup of a NIM database should only be restored to a system with a NIM master fileset that is at the same level or a higher level than the level from which the backup was created.

7.2 Restoring Your SP System

This section covers the procedures to restore the data in the SP system and its different subsystems. The procedures to backup the SP system and any of its subsystems are found in 7.1, "Backing Up Your SP System" on page 139.

7.2.1 Restoring an mksysb Image or a Volume Group Backup

In the following sections we descibe how to restore an mksysb image or a volume group backup.

7.2.1.1 Restoring an mksysb of the CWS

If you have a problem with the CWS and you have a recent backup, you can restore it as follows:

1. Execute the normal procedure used to restore any RS/6000 workstation, as described in Chapter 5. Installing BOS from a system backup is described in *IBM Parallel System Support Programs for AIX: Installation and Migration Guide,* GC23-3898.
2. Run `install_cw`. This script creates the proper node_number entry for the CWS in the ODM. It also executes other functions, as described in 2.10, "The install_cw Command" on page 58.
3. Verify your CWS.

Sometimes you need to recover only one part of a mksysb. For example, if you install a new SP with a PCI CWS and you cannot recover the mksysb that is sent with the SP, because that is taken in a microchannel machine, you will want to recover at least the /spdata directory. To do that, follow these next instructions:

1. Determine the blocksize the tape was set to when the mksys was taken:

   ```
   # cd /tmp
   # tctl -f /dev/rmt0 rewind
   # chdev -l rmt0 -a block_size=512
   # restore -s2 -xqdvf /dev/rmt0.1 ./tapeblksz
   # cat ./tapeblksz
   ```

   Then

2. Set the blocksize of the tape drive accordingly by running the following command:

   ```
   # chdev -l rmt0 -a block_size=[number in the ./tapeblksz file]
   ```

3. Restore the files or directories that you want by executing the following commands:

```
# cd /
# tctl -f /dev/rmt0 rewind
# restore -s4 -xqdvf /dev/rmt0.1 ./< path >
```

7.2.1.2 Restoring a mksysb of a Node

The procedure that is used to restore a mksysb image to a node is the same as that used for installing a node; for a detailed description see 5.3.1, “Method 1: Applying AIX PTFs Using mksysb Install Method” on page 106.

To define your install image in an PSSP 3.1 environment you need to run the following commands:

```
# spchvgobj -r rootvg -i <image name> -l <node_number>
# spbootins -r install -l <node_number>
```

Example:

To restore node 10 with an image called sp4n10.img:

```
# spchvgobj -r rootvg -i image.sp4n10.img -l 10
# spbootins -r install -l 10
```

You can verify the environment as shown in the following command:

```
# splstdata -b -l 10
                List Node Boot/Install Information

node#        hostname  hdw_enet_addr srvr     response           install_disk
     last_install_image   last_install_time  next_install_image lppsource_name
             pssp_ver        selected_vg
-------------------------------------------------------------------------------
   10 sp4n10            10005AFA159D    0      install                hdisk0
        bos.obj.ssp.432 Mon_Apr_12_15:44:09       sp4n10.img          aix432
             PSSP-3.1            rootvg
```

Check the fields *response* and *next_install_image*.

Now you should net boot the node to restore the correct image.

You can restore this image in a node different from the original node without worrying about the number of the node and its specific configuration. That configuration is restored in the customization face of the installation process.

7.2.1.3 The Restore Volume Group Command

The `restvg` command restores the user volume group and all its containers and files, as specified in the /tmp/vgdata/vgname/vgname.*data* file (where vgname is the name of the volume group) contained within the backup image created by the savevg command. Refer to 7.1.1.4, “Savevg” on page 142 to learn how to use this command.

To restore a specific volume group using SMIT, you can do the following command:

```
# smitty restvg
```

Enter the name of the file or device that contains a backup taken with the savevg command.

Enter the physical volume names for that vg (optional)

Press Enter.

Using the command line, type the following:

```
# /usr/bin/restvg -q -f '/temporal/jorgevg.backup'
```

7.2.2 Restoring the SDR

The restore of the SDR is accomplished by using the SDRRestore command. This command is actually a script in the /usr/lpp/ssp/bin directory that deletes the information of the existing SDR, stops and removes the sdr daemons and subsystems, restores the backup taken with the SDRArchive command which is in tar format, and recreates and restarts the subsystems and daemons needed.

You can restore the SDR backup either through SMIT or by using the command line, as follows.

If you use SMIT, type:

```
# smitty syspar
```

Select **Restore System Partition Configuration.**

Select the file you want to restore using F4.

Press Enter.

This way you restore the SDR and also the corresponding partition-sensitive subsystems.

Using the command line:

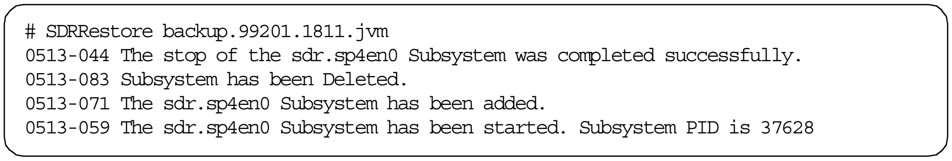
You can use one of two commands:

```
   # SDRRestore <backup file name>
```

or

```
   # sprestore_config <backup file name>
```

The `SDRRestore` command only stops and restarts the sdr subsystem and restores the tar contained in <backup file name>. As an example, Figure 24 shows the output of the execution of the `SDRRestore` command of a file called backup.99201.1811.



```
# SDRRestore backup.99201.1811.jvm
0513-044 The stop of the sdr.sp4en0 Subsystem was completed successfully.
0513-083 Subsystem has been Deleted.
0513-071 The sdr.sp4en0 Subsystem has been added.
0513-059 The sdr.sp4en0 Subsystem has been started. Subsystem PID is 37628
```

Figure 24. Restore of the SDR Using SDRRestore

Alternatively, `sprestore_config` not only restores the SDR, but also restores all partition-sensitive subsystems. This is actually the command executed when you restore using SMIT (see Figure 25).

```
# sprestore_config backup.99201.1811.jvm
0513-044 The stop of the sdr.sp4en0 Subsystem was completed successfully.
0513-083 Subsystem has been Deleted.
0513-071 The sdr.sp4en0 Subsystem has been added.
0513-059 The sdr.sp4en0 Subsystem has been started. Subsystem PID is 17934.
stopping "hr.sp4en0"
0513-044 The stop of the hr.sp4en0 Subsystem was completed successfully.
removing "hr.sp4en0"
0513-083 Subsystem has been Deleted.
0513-071 The hats.sp4en0 Subsystem has been added.
0513-071 The hags.sp4en0 Subsystem has been added.
0513-071 The hagsglsm.sp4en0 Subsystem has been added.
0513-071 The haem.sp4en0 Subsystem has been added.
0513-071 The haemaixos.sp4en0 Subsystem has been added.
Added 0 objects to class EM_Resource_Variable
Added 0 objects to class EM_Structured_Byte_String
Added 0 objects to class EM_Resource_ID
Added 0 objects to class EM_Resource_Class
Added 0 objects to class EM_Resource_Monitor
making SRC object "hr.sp4en0"
0513-071 The hr.sp4en0 Subsystem has been added.
0513-071 The pman.sp4en0 Subsystem has been added.
0513-071 The pmanrm.sp4en0 Subsystem has been added.
0513-071 The Emonitor.sp4en0 Subsystem has been added.
0513-059 The hats.sp4en0 Subsystem has been started. Subsystem PID is 38712.
0513-059 The hags.sp4en0 Subsystem has been started. Subsystem PID is 22996.
0513-059 The hagsglsm.sp4en0 Subsystem has been started. Subsystem PID is 25742.
0513-059 The haem.sp4en0 Subsystem has been started. Subsystem PID is 20558.
0513-059 The haemaixos.sp4en0 Subsystem has been started. Subsystem PID is 24642.
0513-059 The hr.sp4en0 Subsystem has been started. Subsystem PID is 35508.
0513-059 The pman.sp4en0 Subsystem has been started. Subsystem PID is 24880.
0513-059 The pmanrm.sp4en0 Subsystem has been started. Subsystem PID is 40450.
0513-059 The sp_configd Subsystem has been started. Subsystem PID is 18412.
```

Figure 25. Restore of the SDR Using sprestore_config

7.2.3 Restoring the Kerberos Database

If the kerberos database becomes corrupted, it may be necessary to restore it. If you do not have a backup taken with the command `kdb_util dump`, you might have to recreate the database using the `setup_authent` command. If you do have that backup, then you can use the following procedure to restore your database:

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

The realm name will be the same as the domain name, but converted to uppercase.

```
# kstash
# kdb_util load <filename of backup>
# chitab "kerb:2:once:/usr/lpp/ssp/kerberos/etc/kerberos"
# chitab "kadm:2:once:/usr/lpp/ssp/kerberos/etc/kadmind -n"
# telinit 2
# startsrc -s hardmon
# kinit root.admin
```

If you do not have a good backup of the kerberos database, and you need to rebuild it, refer to *RS/6000 SP: Problem Determination Guide,* SG24-4778 for a detailed procedure to reconstruct the database.

7.2.4 Restoring the NIM Database

To restore the NIM database, you can use SMIT or the command line.

If you use SMIT, type:

```
# smitty nim_restore_db
```

Then select the device or file to restore.

When we tried to restore using SMIT, we received error messages because the system asked us to unconfigure the NIM database, but in some cases the system cannot do that and returns errors. It is easier to restore the backup manually. If you use the SMIT menu to backup the database, go to / and execute:

```
# tar -xvf /etc/objrepos/nimdb.backup
```

This command will restore the files of the NIM database immediately.

7.3 Remote Backups and Restores

When you want to back up files or directories in nodes which do not have tape drives directly attached, you have three options:

1. Do a local backup using any command (`tar, dd, cpio, backup`, and so forth) and send the file using `rcp,pcp,ftp` or any other command.
2. Use a specific tool (such as ADSM) to send your data to a backup server.
3. Create a backup on a remote system which has a tape connected; you can use systems like the CWS, another node, or any workstation.

The first option can be good if you have enough space in your local node and have a slow network; by using this option, you can take the backup fast, compress the file, and send it to the other machine.

If you want to take a backup directly to a file or tape in other machine, you can use the `tar` and `dd` commands in the following way:

```
# tar -cvf - <path> | rsh <remote host> dd of= <remote tape or file> \
bs=<block size>
```

Remember to use the dash symbol (-) before the path, and also remember that the block size depends of current tape device settings. For example, take a backup of the /tmp filesystem to the tape /dev/rmt0 connected to a machine called sp4en0 and defined with a block size of 1024 using the `dsh` command, you must execute:

```
# tar -cvf - /tmp | dsh -w sp4en0 dd of=/dev/rmt0 bs=1024
```

However, running this command, you can experience some problems. For example, if you have a block size that does not match the one in the remote tape, you will receive a response similar to the following:

```
# tar -cvf - ./usr/lib/u* |rsh sp4en0 dd of=/dev/rmt0 bs=512
a ./usr/lib/unixtomh 19 blocks.
a ./usr/lib/uucp
a ./usr/lib/uucp/Devices symbolic link to /etc/uucp/Devices.
a ./usr/lib/uucp/Dialers symbolic link to /etc/uucp/Dialers.
dd: 0511-053 The write failed.
: A system call received a parameter that is not valid.
3+0 records in.
0+0 records out.
```

Also, make sure that you have permission to execute the `remote` command. If you are using kerberos, ensure that you have the proper tickets. If you are using normal remote commands, ensure that you have permission in the /.rhosts file.

To restore a backup taken like this, you must execute the following command:

```
# rsh <remote host> dd if=<remote tape or file> ibs=<block size> |tar \
-xvf - <path>
```

Remember to use the same block size that you used to take the backup.

To create a savevg tape remotely, you must use a pipe; run the following commands:

```
# mknod /tmp/pipe p
# savevg -i -f /tmp/pipe <vgname> &
# dd if=/tmp/pipe |dsh -w <remote host> dd of=<remote tape or file> \
bs=<block size>
```

Then, to restore a savevg, type the following:

```
# mknod /tmp/pipe p
# dsh -w <remote host> dd if=<remote tape or file> bs=<block size> \
</dev/null >/tmp/pipe &
# restvg /tmp/pipe
```

One common requirement when working with the SP2 is the need to install remotely, any third-party applications that require a “local” tape drive to install their product. To accomplish this, assuming that the product is installed with the syntax `installxx <device>`, run the following procedure:

```
# mkfifo /tmp/pipe
# dsh -w sp4en0 <remote host> dd if=<remote tape> bs=<Block size> \
  >/tmp/pipe 2>/dev/null
# installxx /tmp/pipe
```

Chapter 8. Taming Kerberos

Ever since Kerberos was introduced into PSSP software there have been complaints about it. The question is often asked: "Isn't it possible to get rid of this Kerberos stuff on the SP?" The reasons for these complaints are always the same: scripts do not run because of expired Kerberos tickets, the switch is not startable, spmon does not work after changing the IP configuration, distributed shell commands return errors, and so on. As a consequence, many customers stored a /.rhosts file on the nodes just to be sure that remote shell commands run even if Kerberos does not work properly. But one of the reasons for using Kerberos in the first place was to eliminate the need for an /.rhosts file.

"With the SP comes Kerberos and with Kerberos comes trouble." These are the very first words of a service bulletin about Kerberos in 1995. In this chapter we show that this statement is not true and that it can be easy to handle Kerberos as soon as the processes of Kerberos authentication are made clear.

In Greek mythology, Kerberos is the three-headed dog who guards the entrance to Hades. In the following sections, we provide the information, hints and tips you need in order to “tame” Kerberos.

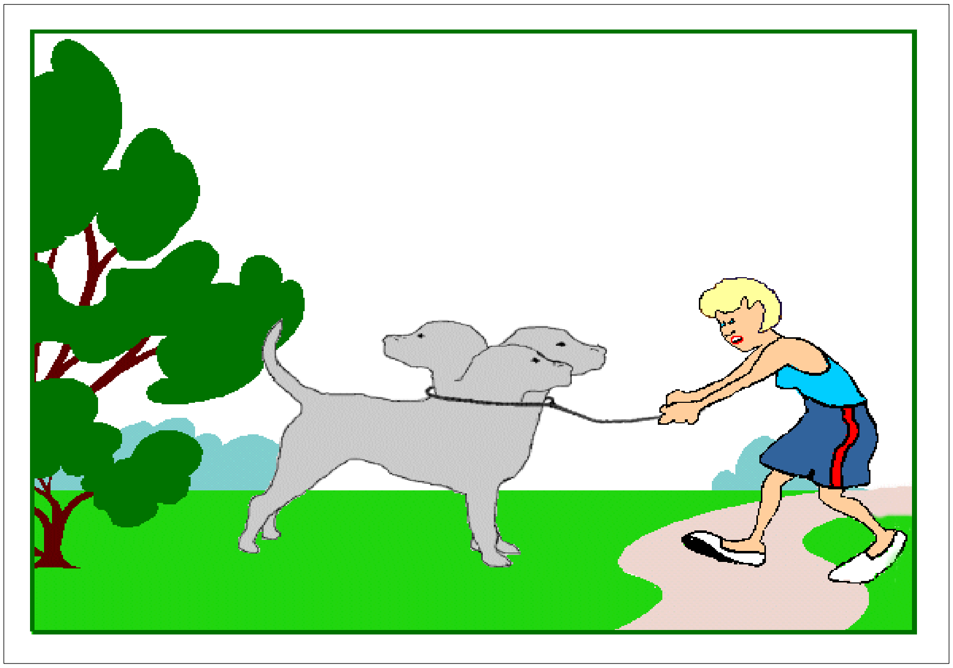


Figure 26. Kerberos as a Domestic Dog

We start with a brief overview of what a default SP Kerberos setup looks like, and explain the necessary Kerberos-related terms. We then approach Kerberos in a chronological way, corresponding to the installation process of an SP. We explain: what does `setup_authent` do? What happens during `setup_server`? Which files are needed where, and when are they distributed?

8.1 The Default SP Kerberos Realm

When you install an RS/6000 SP system with the default settings, your Kerberos setup will look like Figure 27.

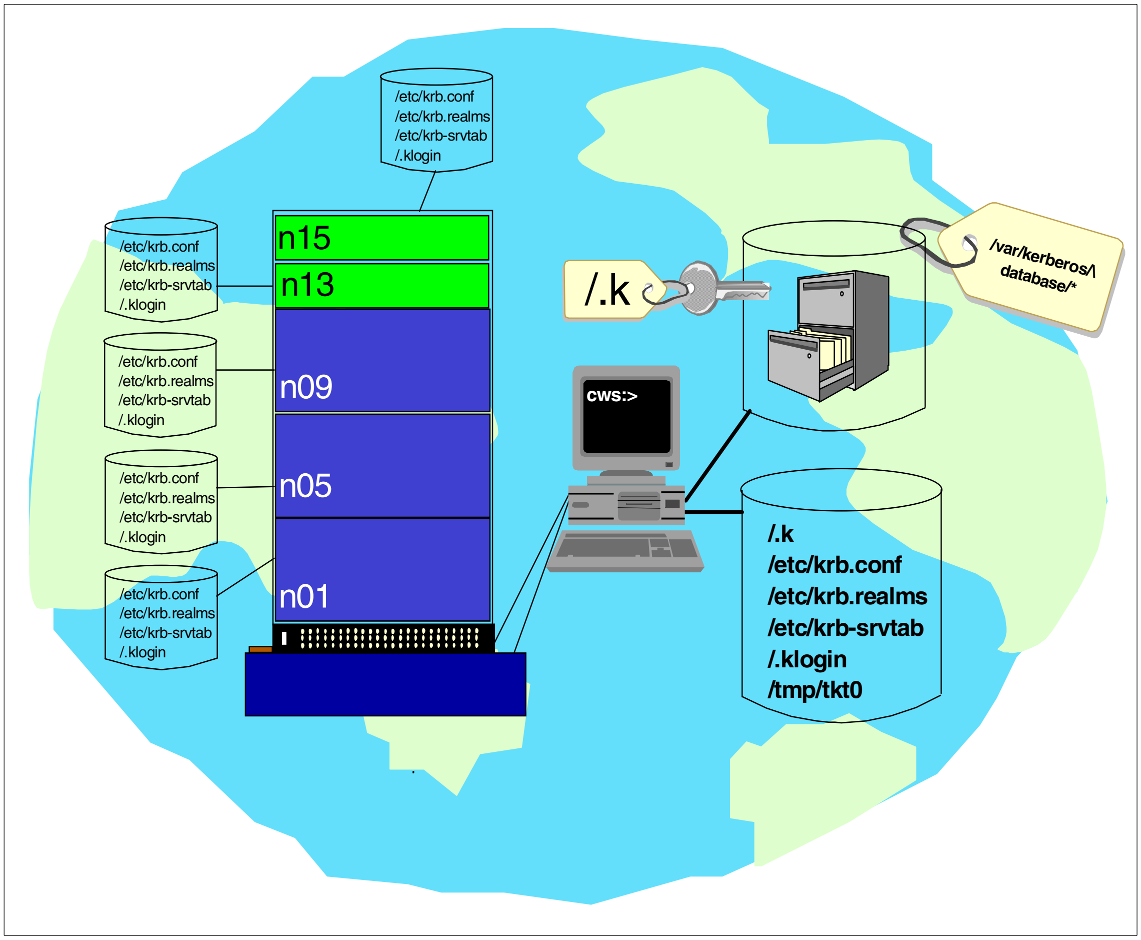


Figure 27. SP Kerberos Realm

The default SP Kerberos environment consists of the CWS as the Kerberos authentication server. All hosts (that means all SP nodes) that are under control of the authentication server belong to its realm. *Realm* in this case is a synonym for kingdom, so the authentication server as the "king" reigns over his kingdom and keeps an eye (or better, an ear) on the communication of all the inhabitants of the kingdom.

On the authentication server, the Kerberos database is stored under the /var/kerberos/database. This database holds the information of all Kerberos users and services and their passwords within this realm. Only the authentication server (that means the CWS) has access to this database because it is encrypted with the so-called Kerberos master key. This master key is stored in the /.k file on the server.

Since there is one central database for one Kerberos realm, all Kerberos user and service names must be unique within this realm. All clients of this Kerberos realm, in our example all SP nodes, have Kerberos configuration files, therefore they know who their authentication server is and which nodes belong to the same realm. As you can see in Figure 27 on page 157, the client nodes have almost all the Kerberos-related files as their authentication server. But some files are unique on every host. Figure 28 shows that the /etc/krb.conf, /etc/krb.realms, and /.klogin files on the clients are just copies of the server files, while the /etc/krb-srvtab file and the /tmp/tkt0 file (if it exists) are unique on each host, although the names are the same.

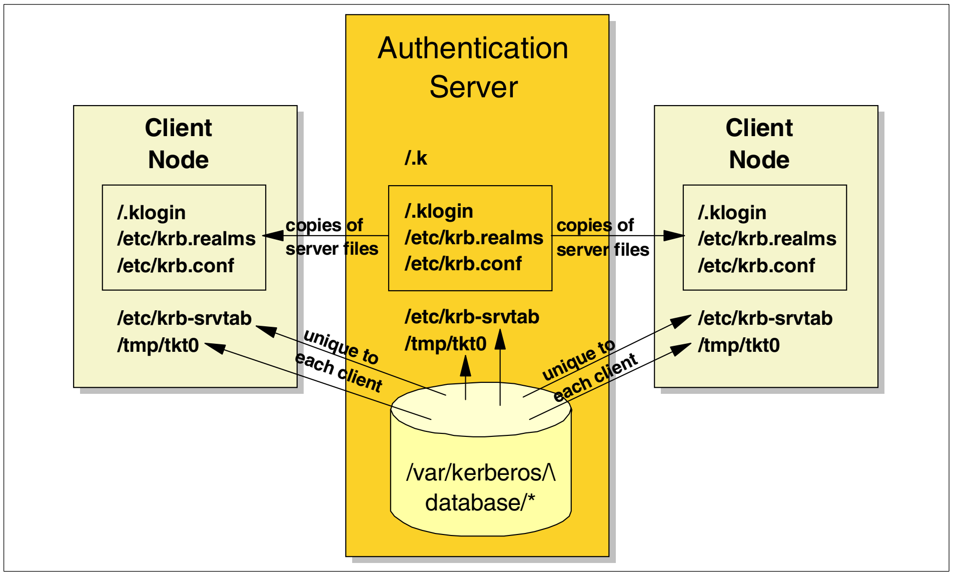


Figure 28. Kerberos Configuration Files

8.2 Kerberos Setup during the Installation of an RS/6000 SP System

Assume we are going to install an RS/6000 SP system. Only the CWS is installed with AIX; the nodes are not yet installed. Note: it is not mandatory that the CWS plays the role of the authentication server. The SP environment can also be included in an existing AFS or CDE cell, for example. However, in most cases the CWS is the authentication server, so this will also be our initial setup.

After the installation of the PSSP code on the CWS, you have to issue `setup_authent`.This script defines the CWS as authentication server, creates the Kerberos database, and asks you for the Kerberos master key, the very first Kerberos principal (root.admin) and some default values. Let us see which files are already created at this point of the installation.

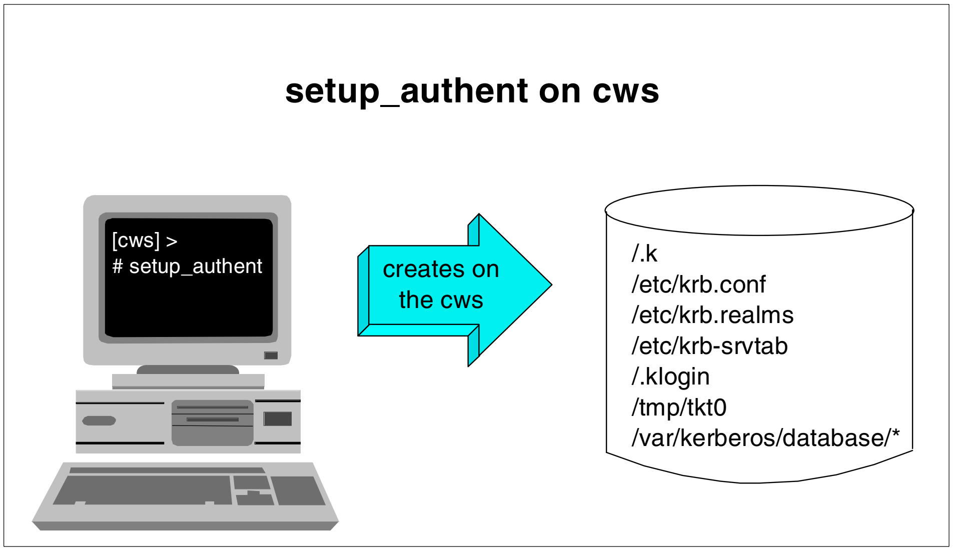


Figure 29. setup_authent on the CWS

8.2.1 The Kerberos Master Key

The /.k file stores the Kerberos master key that has been generated by encrypting the Kerberos master password you have been prompted for during `setup_authent`. This key encrypts not only the whole Kerberos database, but also the ticket-granting-tickets. These tickets are discussed in 8.7, “Tickets and Keys” on page 168. If you remove this /.k file by mistake, you can regenerate it with the `kstash` command.

8.2.2 The Kerberos Configuration File

The /etc/krb.conf file contains information indicating to which Kerberos realm this host belongs and also which host is the authentication server of this realm. Unlike the Kerberos master key, the /etc/krb.conf file is a regular ASCII file.

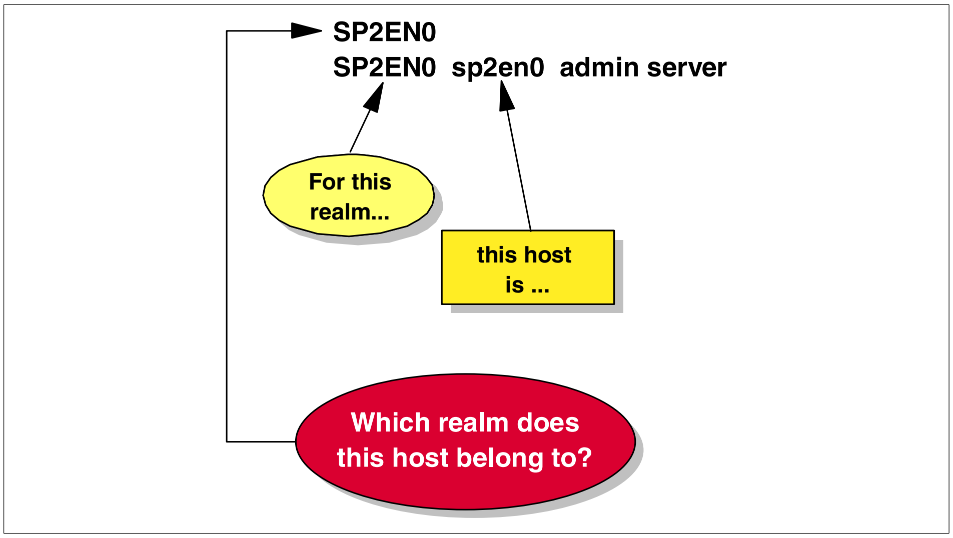

SP2EN0
SP2EN0 sp2en0 admin server
For this realm...
this host is ...
Which realm does this host belong to?

Figure 30. The /etc/krb.file File

As shown in Figure 30, the /etc/krb.conf file has at least two entries. The first line indicates that this host belongs to, for example, the SP2EN0 realm, and the second line gives the information about the authentication server of this realm.

You may ask, who chose the realm name SP2EN0? The `setup_authent` script does the following. First it checks whether a /etc/krb.conf file already exists. If not, `setup_authent` takes the hostname of the CWS and changes it into uppercase to define the realm name. As you can see in Figure 30, the hostname of the CWS is sp2en0, so the realm name became SP2EN0. When you use a domain name server, `setup_authent` uses the domain entry of the /etc/resolv.conf file as the realm name and changes it into uppercase.

Note, however, that you are not forced to use any of these realm names. You can choose the realm name and create your own /etc/krb.conf file in the same syntax. As soon as `setup_authent` detects an existing /etc/krb.conf file, the first entry will then become your realm name.

8.2.3 The Kerberos Realm Configuration

The /etc/krb.realms file is a kind of map file that indicates which network interface (therefore which adapter names) belongs to which realm. Kerberos authentication works with adapter names, therefore you will see an entry for every adapter that is part of the realm. At this stage of the `setup_authent` process, the krb.realms only includes information about the authentication server itself (the CWS), because no node information exists yet. Here is the content of the file after `setup_authent` was executed.

```
sp2en0:/# cat /etc/krb.realms
sp2cw0 SP2EN0
```

In our setup the CWS has a Token Ring adapter with the adapter name sp2cw0 and an Ethernet adapter with the sp2en0 adapter name. The hostname of our CWS is based on the Ethernet adapter (look at the command prompt in the screen output above). Since the primary authentication server is using the hostname, the related adapter name (the ethernet adapter name in our configuration) is not included in the /etc/krb.realms file. Any additional configured adapter, like our Token Ring adapter, will be included in the krb.realms file.

This file will get enlarged with all the adapter names of supported adapters (like Ethernet, Token Ring, Switch adapter, FDDI etc.) from all the nodes during the installation steps following the setup_authent command.

8.3 Kerberos Principals

In a Kerberos environment, users as well as services are called *principals*. You can authenticate yourself to Kerberos either as a user principal or as a service principal.

The complete instance of a principal is composed of these elements:

<principal_name>.<instance>@REALM

Examples: root.admin@SP2EN0, anna.kerb@SP2EN0

The very first Kerberos user principal that you define (usually root.admin) must have the instance *admin.* This instance is needed to do administrative tasks on the Kerberos server. This Kerberos principal must be logged in as UNIX user root. For further user principals, the instance name is freely choosable, but it must not be created before you define the principal.

In an SP environment two service principals are available:

- rcmd.*<instance>@REALM* (on CWS and on each node)
- hardmon.*<instance>@REALM* (only on the CWS)

The *instance name* of a service principal is the adapter name on which this service is provided. The rcmd service-principal provides the Kerberos remote commands. Since you want to be allowed to set up Kerberized remote commands over each network in your SP, you have as many rcmd principals as the number of adapters in your SP configuration.

When you are allowed to use the hardmon service on the CWS, you can talk to the hardmon daemon that monitors the serial link to the SP frame. Without hardmon service, you are not able to run any command that uses the serial line: these commands are `spmon`, `s1term`, `hmcmds`, and `hmmon`. That is why you cannot run `spmon` commands without a working Kerberos environment. By default only root.admin is allowed to use the hardmon services, but you can change this configuration; see 8.7.7, “Hardmon Access Control List - hmacls File” on page 177, for more information.

8.3.1 The /.klogin File

If you want to know which principal is allowed to use the Kerberos remote commands and services of a node, look in the /.klogin file. This file contains a list of all principals that may use a service. Even if you are authenticated as a Kerberos principal and you try to contact a remote host using Kerberized commands, this /.klogin file will be checked after a lot of other authentication actions. Usually you are authenticated as a principal. If your principal name is not included in the .klogin file and you are trying to run a remote command, the execution will be refused. The following screen shows what this file looks like.

```
sp2en0:/# cat /.klogin
root.admin@SP2EN0
anna.sap@SP2EN0
rcmd.sp2en0@SP2EN0
rcmd.sp2n01@SP2EN0
rcmd.sp2n05@SP2EN0
rcmd.sp2n06@SP2EN0
rcmd.sp2n07@SP2EN0
rcmd.sp2n08@SP2EN0
rcmd.sp2n09@SP2EN0
rcmd.sp2n10@SP2EN0
rcmd.sp2n11@SP2EN0
rcmd.sp2n12@SP2EN0
rcmd.sp2n13@SP2EN0
rcmd.sp2n14@SP2EN0
rcmd.sp2n15@SP2EN0
```

8.4 Kerberos Daemons

There are two daemons running on the authentication server which belong the Kerberos system, the kadmin daemon and the kerberos daemon. This is shown in the following screen.

```
sp2en0:/# ps -ef | grep kerb
root 21614 22670   0 14:40:38  pts/6  0:00 grep kerb
root 27664  7744   0 10:38:29      -  0:00 /usr/lpp/ssp/kerberos/etc/kadmind -n
root 33850  7744   0 10:38:25      -  0:00 /usr/lpp/ssp/kerberos/etc/kerberos
```

The kerberos daemon is used for the authentication of Kerberos users. The kadmind is only used during administrative tasks on the Kerberos database (for example adding principals, changing passwords, and so on). It happens that using remote commands like dsh work fine, but issuing a kadmin command hangs; no input prompt returns. In such cases, make sure both daemons are running. You will not notice the death of the kadmind daemon as long as you are only asking for tickets and services.

These daemons are started from the /etc/inittab. Also, starting with PSSP 2.3, these daemons are managed by the system resource controller. You can start and stop them as follows:

```
# stopsrc -s kadmin
# startsrc -s kadmin
```

A typical screen output of such a command sequence in shown in the following screen.

```
sp2en0:/# stopsrc -s kadmind
0513-044 The stop of the kadmind Subsystem was completed successfully.
sp2en0:/# startsrc -s kerberos
0513-059 The kerberos Subsystem has been started. Subsystem PID is 35738.
sp2en0:/# lssrc -a |grep kerb
 rpc.lockd       nfs             11096   active
 kerberos                        35738   active
 kadmind                                 inoperative
```

Before PSSP 2.3 version, in order to stop kerberos and kadmind, you had to change the respawn value in the /etc/inittab file of these daemons to off and then kill the corresponding processes.

8.5 setup_authent, setup_server and the Kerberos Database

All Kerberos principals, both users and services, are stored in the Kerberos database. But did you ever define a Kerberos principal except root.admin during an SP installation? All service principals (hardmon and rcmds) are defined automatically either by `setup_authent` or by `setup_server`.

In our example, `setup_authent` did run, but not `setup_server`, so what is the content of the Kerberos database at this stage? Let us look into the database by issuing the `lskp` command.

```
sp2en0:/# lskp
K.M                 tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
changepw.kerberos   tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
default             tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
hardmon.sp2cw0      tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
hardmon.sp2en0      tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
krbtgt.SP2EN0       tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2cw0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2en0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
root.admin          tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
```

As shown in the preceding screen, you can see the root.admin user principal, hardmon service and rcmd principals for the CWS.

After filling up the SDR with node information, you execute `setup_server` for the very first time. One wrapper of `setup_server`, called `setup_CWS`, scans the SDR for the adapter names for all nodes. For every adapter name it then adds one rcmd principal, and sets a password for it. After running `setup_CWS`, the

Kerberos database contains many more entries, as shown in the following screen.

```
sp2en0:/# lskp | pg
K.M                 tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
changepw.kerberos   tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
default             tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
hardmon.sp2en0      tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
krbtgt.SP2EN0       tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2cw0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2en0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css01       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css05       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css06       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css07       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css08       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css09       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css10       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css11       tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n01         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n05         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n06         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n07         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n08         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n09         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n10         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2n11         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
root.admin          tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
```

Now there is one rcmd principal defined for each adapter in the SP system. As already mentioned, this service principal provides the Kerberized remote commands.

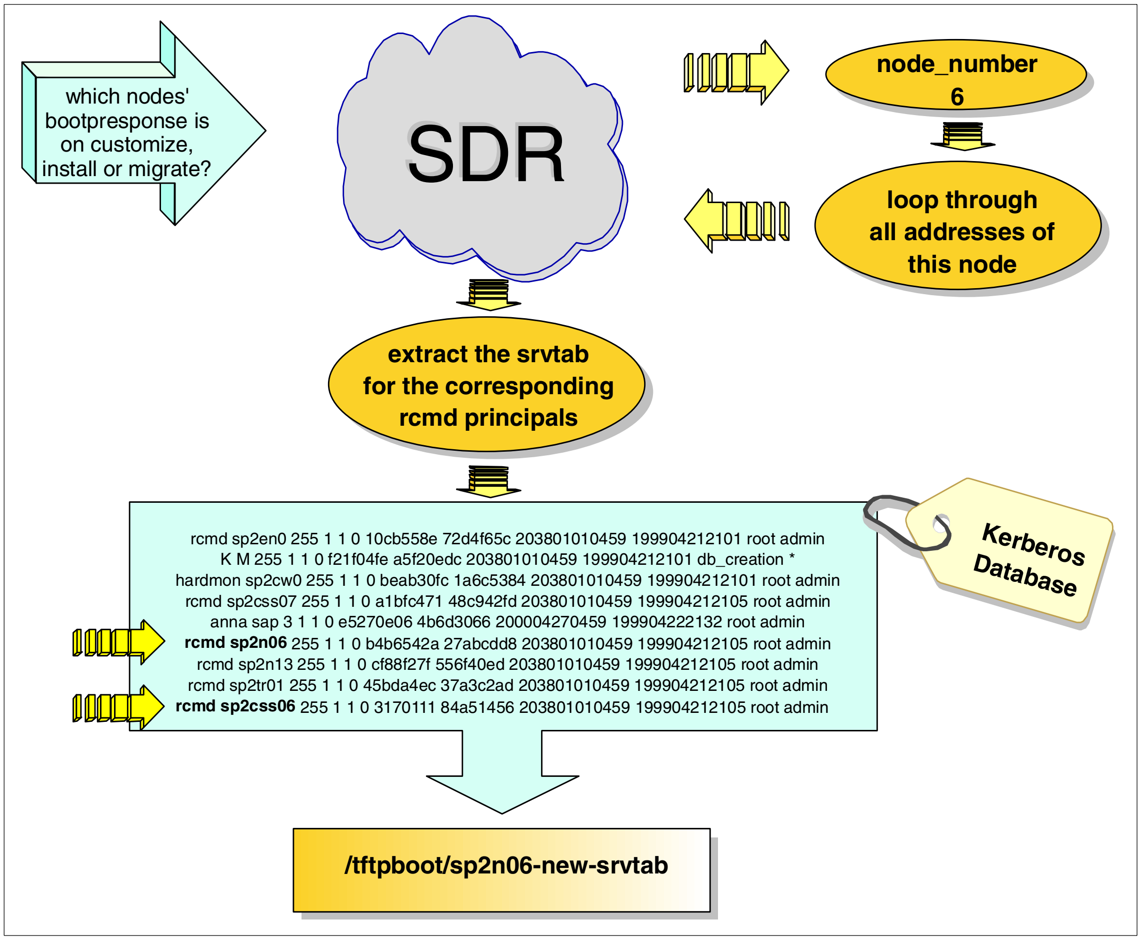
The /etc/krb.realms file has also been filled up with all these new adapters. This file indicates which adapter belongs to which realm. The example shows only the first part of it.

```
sp2en0:/# cat /etc/krb.realms | pg
sp2cw0 SP2EN0
sp2n01 SP2EN0
sp2tr01 SP2EN0
sp2css01 SP2EN0
sp2n05 SP2EN0
sp2css05 SP2EN0
sp2n06 SP2EN0
sp2css06 SP2EN0
sp2n07 SP2EN0
sp2css07 SP2EN0
sp2n08 SP2EN0
sp2css08 SP2EN0
^C
```

8.6 The krb-srvtab File

The rcmd principals have been defined and a random password has been set for them, as you set it for the root.admin principal during the installation of the CWS. These rcmd passwords are all stored in the Kerberos database and also in the /etc/krb-srvtab files.

While `create_krb_files` (a wrapper of setup_server) is running, these srvtab files are created for each node that has a bootp_response value of install, customize or migrate in the SDR. These srvtabs are node-dependent and contain the encrypted passwords for the rcmd principals of each node. The different srvtab files are extracted from the Kerberos database and at first stored under /tftpboot/<node_name>-new-srvtab on the CWS. In Figure 31 on page 167, you can follow the creation of the new krb srvtab files on the CWS for the SP nodes:


which nodes' bootpresponse is on customize, install or migrate?
SDR
node_number 6
loop through all addresses of this node
extract the srvtab for the corresponding rcmd principals
Kerberos Database
/tftpboot/sp2n06-new-srvtab

```
rcmd sp2en0 255 1 1 0 10cb558e 72d4f65c 203801010459 199904212101 root admin
K M 255 1 1 0 f21f04fe a5f20edc 203801010459 199904212101 db_creation *
hardmon sp2cw0 255 1 1 0 beab30fc 1a6c5384 203801010459 199904212101 root admin
rcmd sp2css07 255 1 1 0 a1bfc471 48c942fd 203801010459 199904212105 root admin
anna sap 3 1 1 0 e5270e06 4b6d3066 200004270459 199904222132 root admin
rcmd sp2n06 255 1 1 0 b4b6542a 27abcdd8 203801010459 199904212105 root admin
rcmd sp2n13 255 1 1 0 cf88f27f 556f40ed 203801010459 199904212105 root admin
rcmd sp2tr01 255 1 1 0 45bda4ec 37a3c2ad 203801010459 199904212105 root admin
rcmd sp2css06 255 1 1 0 3170111 84a51456 203801010459 199904212105 root admin
```

Figure 31. The create_krb_files Wrapper

At the end of a node installation, the customize script pssp_script is running on the node. This script fetches the Kerberos-related files and the appropriate new srvtab and moves it to /etc/krb-srvtab. Though the srvtab, this file actually is not readable, as shown in the srvtab file of node07:

```
p2n07:> cat /etc/krb-srvtab
rcmdsp2css07SP2EN0½Ïuî¯LŠrcmdsp2n07SP2EN0´bøÄ#
```

However, it is possible to identify the principal strings: rcmd followed by the realm string. Our example output in the previous screen shows the passwords for the principals rcmd.sp2css07@SP2EN0 and rcmd.sp2n07@SP2EN0.

After the installation of the whole SP, each SP node has its own /etc/krb-srvtab file.

8.7 Tickets and Keys

There are two kinds of tickets used in an Kerberos environment:

- Ticket-granting-ticket
- Service-ticket

To point out the difference between these two kinds of tickets, let us imagine a trip to Disney World. At the entrance you have to buy a ticket just to enter, and this ticket allows you to stay inside for a defined period of time. This initial ticket can be compared to the Kerberos ticket-granting-ticket which has a specified lifetime. It only gives you permission to use Kerberos services in this realm. But you do not enter Disney World (or a Kerberos realm) just to get inside and then stay in only one place; instead, you are going to have some fun and visit several shows.

If Disney World was organized like a Kerberos realm, there would be a central ticket counter. In a Kerberos realm, this ticket counter (called the Ticket-Granting-Server) is located in the authentication server and it already issued you the ticket-granting-ticket. To get access to one of the Disney shows or even better -- to one of the Kerberos services, you have to show your ticket-granting-ticket at the central ticket counter. It will be checked and if it is accepted, you will obtain a ticket (in Kerberos terminology, a *service ticket)* to visit the show -- or use Kerberos remote services. Without a ticket-granting-ticket, you will not get any access.

Now keys, as we know, are used both to lock something and to open it again. If two people share the same key, one can lock and the other can reopen. In the Kerberos environment, we use two different keys:

- Session Key
- Private Key

While a Session Key is generated randomly and used for communication between two parties, the private key is generated based on the principals' password plus a timestamp.

How are these tickets and keys distributed and used without sending any unencrypted passwords over the network? If you are interested in the whole theory of Kerberos encryption, refer to *Inside the RS/6000 SP,* SG24-5145.

8.7.1 Ticket-Granting-Ticket

The last part of the `setup_authent` command invokes the `kinit` command for the first principal you have defined, usually root.admin. This `kinit root.admin` prompts you again for the password of this principal, then provides you with a ticket-granting-ticket. Since this ticket-granting-ticket has by default a lifetime of 30 days, you have to get a new one after expiration. To obtain a new ticket-granting-ticket for root.admin, just issue `kinit root.admin` at the command prompt. If you are interested in the values of your current ticket, `klist` gives you the information needed, as shown.

```
sp2en0:/# kinit root.admin
Kerberos Initialization for "root.admin"
Password:
sp2en0:/# klist
Ticket file:    /tmp/tkt0
Principal:      root.admin@SP2EN0

  Issued          Expires         Principal
Apr 16 10:58:08  May 15 10:58:08  krbtgt.SP2EN0@SP2EN0
```

To say `kinit <principal.instance>` has been compared to logging on to Kerberos, while the Kerberos identity you get is dependent on your UNIX identity. In our example, the UNIX root user is authenticated as the Kerberos root.admin principal. This combination of two identities is kept in your ticket-granting-ticket.

The first line of the `klist` output always points to your ticket cache file that stores your ticket; therefore, it will survive a reboot of the machine (see 8.7.4, “The Ticket Cache File” on page 174). The second line indicates the owner of this ticket, and the first line of the next block represents the ticket-granting-ticket itself.

8.7.2 Contents and Purpose of the Ticket-Granting-Ticket (tgt)

It does not matter whether you get a root.admin ticket-granting-ticket (tgt) on the host that plays the role of the authentication server, or one of your client nodes. The rule is that a password must not pass the network unencrypted.

In the authentication server itself exists an instance, the so-called Ticket-Granting-Server (TGS), that generates the tickets. Assume you are going to get a tgt for root.admin on n07. Due to the fact that all nodes belonging to a realm have a copy of the /etc/krb.conf file that points to the authentication server, the `kinit` command detects which server has to be contacted. Figure 32 shows the process of getting a tgt and the contents of that tgt:

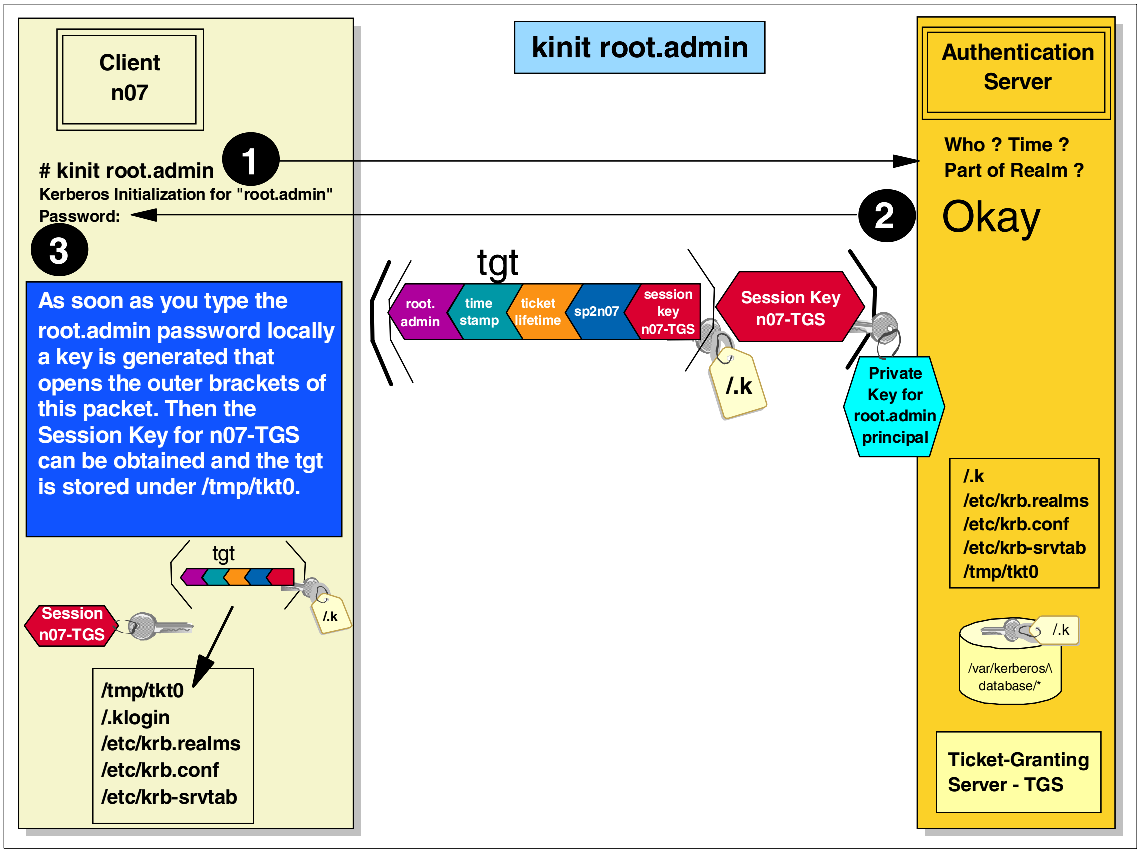


Figure 32. The kinit Procedure

Step 1: The client n07 contacts the authentication server asking for a ticket-granting-ticket for the root.admin principal.

Step 2: The authentication server verifies whether this client is part of the realm and creates a packet including a tgt encrypted with the master key (/.k), so that only the authentication server itself is able to open it. This tgt contains the client’s name, the name of the Ticket-Granting-Server, a timestamp, the ticket lifetime, the client’s IP address, and the Session Key for n07-TGS. Furthermore, a copy of this Session Key n07-TGS is appended to the tgt. The whole packet is encrypted with the root.admin's private key, which is a key that is generated based on the root.admin password plus a timestamp.

Since the root.admin password is stored in the Kerberos database, the authentication server (or TGS) can generate the key locally. Now the packet is sent back to the client.

Step 3: When this packet arrives at the client node, the password prompt appears. The root.admin password is issued locally and the Kerberos code on the client generates a key as well, based on that password. If the password was correct, the generated key is able to open the packet. Only the session key contained in the outer brackets is usable by the client (node n07) for further communication. The ticket-granting-ticket itself is always a kind of "black box" for the client. Only the authentication server is able to open it with its master key (/.k). This tgt is now stored under /tmp/tkt0 on client n07.

8.7.3 Asking for a Remote Service

Now n07 holds a tgt for the root.admin principal. But to use the remote services of another node he needs an additional service ticket for that node. He has to contact the authentication server again, showing his tgt and asking for a service ticket for the remote node.

Let us assume n07 asks for a remote service of n08; n07 wants to set up a remote shell command on n08. We divide this procedure into two steps, first the communication with the authentication server and then the direct communication between n07 and n08.

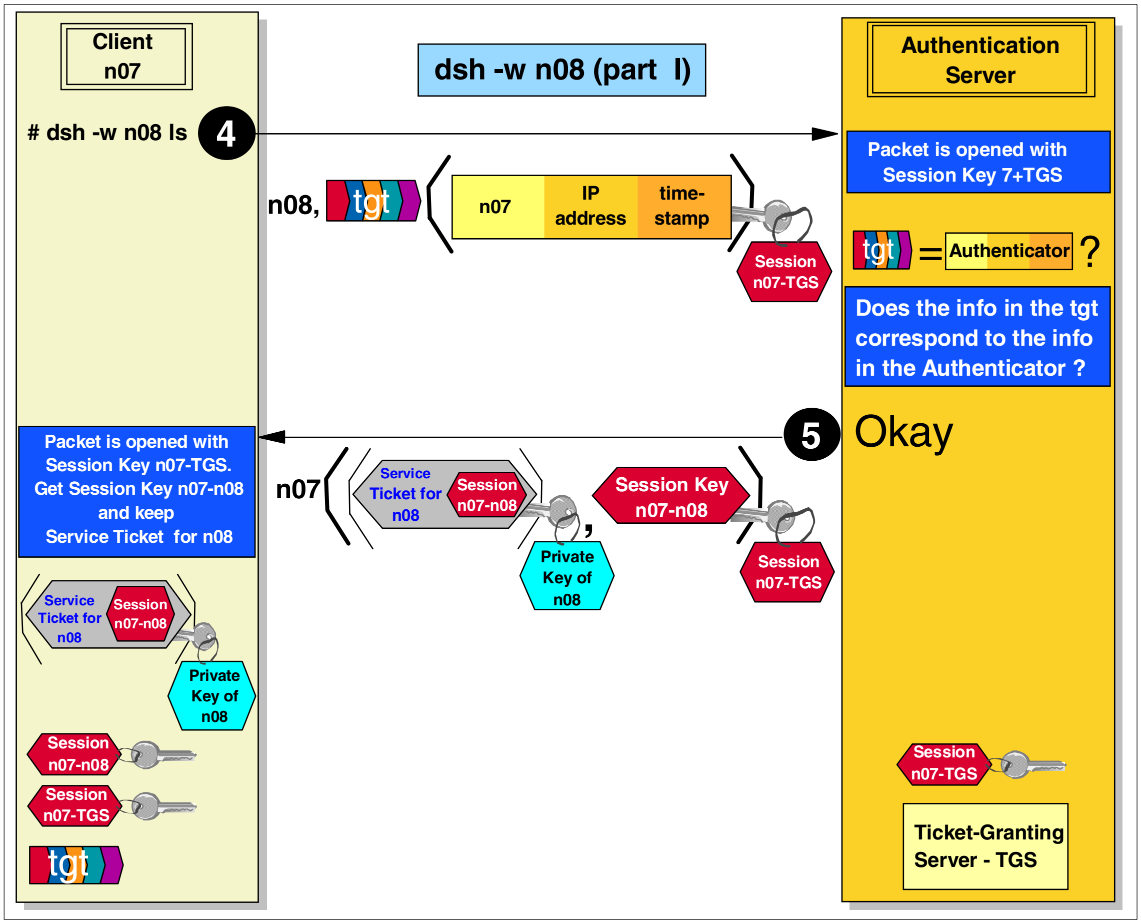


Figure 33. Requesting a Service Ticket

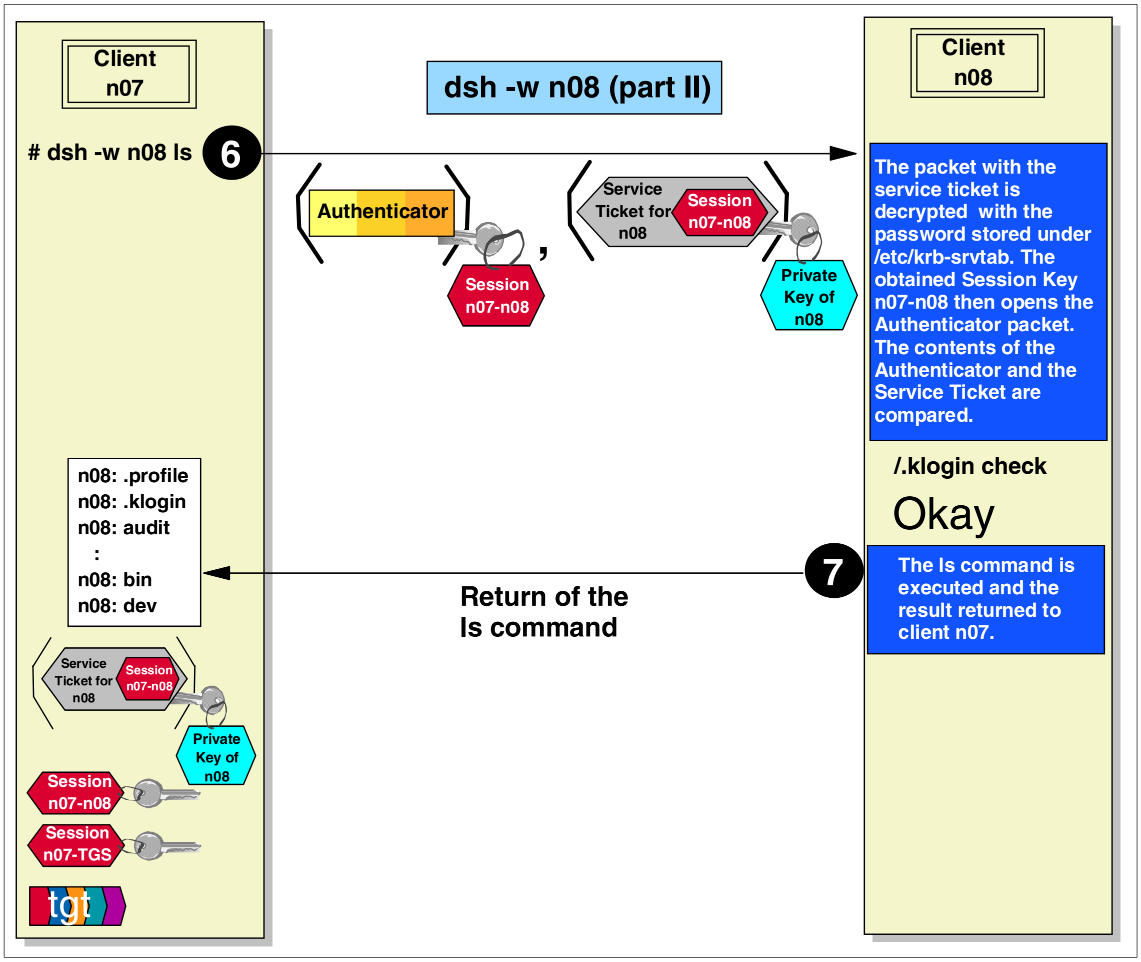
Step 4: Client n07 wants to set up an `ls` command on n08 via remote shell. Since n07 has no service ticket for n08 yet, the authentication server has to be contacted first. n07 wraps a packet with the following contents: n08, the host that wants to be contacted, the ticket-granting-ticket, and an authenticator storing the name and IP-address of n07, plus a timestamp. This authenticator is encrypted with the Session Key n07-TGS. This packet is sent to the authentication server.

Unlike the tickets, an authenticator can only be used once.

The authentication server opens this packet with its Session Key n07-TGS and opens the tgt with its master key. The content of the tgt is compared to the content of the authenticator to insure that n07 is the valid owner of the tgt. If the check returns okay, the next step (step 5) will be executed.

Step 5: The authentication server returns a packet to n07. The first part contains the Service Ticket for n08 including the Session Key for n07 and n08 (n07-n08). This part is encrypted with the private key of n08 so that only n08 is able to open it. Then the same Session Key n07-n08 is appended and the whole packet encrypted again with the Session Key n07-TGS. 07 opens the packet with its Session Key n07-TGS and gets the Session Key n07-n08 and an encrypted packet containing the Service Ticket for n08.

In Figure 34 you can follow part II of the `dsh` command:



The packet with the service ticket is decrypted with the password stored under /etc/krb-srvtab. The obtained Session Key n07-n08 then opens the Authenticator packet. The contents of the Authenticator and the Service Ticket are compared.

The ls command is executed and the result returned to client n07.

Figure 34. Asking for Remote Services

Step 6: Client n07 now has the Service Ticket and the integrated Session Key. Since all this is encrypted with the private key of n08, it can only be opened by n08 with the appropriate password stored in /etc/krb-srvtab. Now the Service Ticket and the Session Key n07-n08 are obtained. With the just-extracted Session Key n07-n08, the Authenticator can be decrypted. The contents of the authenticator are compared with the contents of the Service Ticket.

Step 7: If the comparison was successful and the principal root.admin@REALM (who is asking for the remote service) is included in the /.klogin, the `ls` command will be executed. The screen output of the command will be returned to n07. If one of these two conditions was not met the remote service for n07 will be refused.

8.7.4 The Ticket Cache File

The ticket cache file stores all tickets that a Kerberos principal has obtained, tgts as well as service tickets. The service tickets live as long as the ticket-granting-ticket. As soon as service tickets for all clients are appended to that file, the authentication server is not longer contacted when a remote service is asked for since all the tickets are available and the authenticator is created locally.

In general the Kerberos ticket cache files are stored in the /tmp directory. The file name tkt0 by default is composed of tkt followed by the UID of the UNIX user that asked for this ticket-granting-ticket. Since the root user has the UID 0, the ticket cache file's name is /tmp/tkt0. However, you can name this file whatever you want since the correlation of UNIX UID and Kerberos principal is integrated in this file. Having the UID as the ending of the file just facilitates the identification for you.

You may want to store the ticket files in another location. This can be done with the KRBTKT environment variable in the ~/.profile; for example:

```
# export KRBTKFILE=~/tkt$LOGIN
```

This command will name the tgt for UNIX user anna to be /home/anna/tktanna. Wherever you decide to store the tickets, insure that each user who asks for a ticket is allowed to write to this directory; that is the reason why it is stored in /tmp by default.

The fact that the ticket-granting-ticket is a combination of the UNIX and Kerberos identity makes it possible that several UNIX users can log on or authenticate as the same Kerberos principal. The principal name is totally independent from the UNIX user name. It may be the same but it does not need be. Let us look at an example.

Assume we have two UNIX users on the system and a Kerberos user principal called anna.sap (refer to 8.8.2, “Make a Kerberos Principal” on page 181 for information about how a new Kerberos principal is created).

```
p2en0:/# lsuser anna,heiner | awk '{print $1" "$2}'
anna id=203
heiner id=209
```

Both anna and heiner can authenticate themselves as Kerberos principal anna.sap (if both know the password) without any collisions, as shown in the following screens:

```
sp2en0:/home/anna $ id
uid=203(anna) gid=1(staff)
sp2en0:/home/anna $ kinit anna.sap
Kerberos Initialization for "anna.sap"
Password:
sp2en0:/home/anna $ klist
Ticket file:    /tmp/tkt203
Principal:      anna.sap@SP2EN0

  Issued           Expires          Principal
Apr 16 16:13:36  May 15 16:13:36  krbtgt.SP2EN0@SP2EN0
```

```
sp2en0:/home/heiner $ id
uid=209(heiner) gid=1(staff)
sp2en0:/home/heiner $ kinit anna.sap
Kerberos Initialization for "anna.sap"
Password:
sp2en0:/home/heiner $ klist
Ticket file:    /tmp/tkt209
Principal:      anna.sap@SP2EN0

  Issued           Expires          Principal
Apr 16 16:17:02  May 15 16:17:02  krbtgt.SP2EN0@SP2EN0
```

Since anna and heiner have different ticket cache files, they can share one Kerberos principal. You may ask: "For what purpose?”

8.7.5 Authentication and Authorization

Say we have an SP system with 12 nodes. All the nodes belong to the same Kerberos realm. You provide 8 nodes to login for your UNIX users. Each UNIX user is known on each node and you want to allow remote shell commands between the nodes. You have the choice of creating a /etc/hosts.equiv file or ~/.rhosts files, or else the users are prompted for their passwords when using `rexec`. All these possibilities are more or less unsecure, especially the .rhosts file. As shown in the last section (8.7.4, “The Ticket Cache File” on page 174), several UNIX users can share one Kerberos principal because the combination of UNIX and Kerberos identity makes the Kerberos ticket unique.

Kerberos does the authentication and permits or denies (by providing you a ticket or not) *sending* remote shell commands to the hosts belonging to the same realm. The authorization (meaning the permission to read, write or execute a file) is still done by the remote UNIX system by checking the UNIX permissions. For instance, whether or not anna is allowed to copy a file to a remote machine is under the control of the remote UNIX system. If anna is not known on the remote host, the `rcp` command will be refused even if the Kerberos authentication was okay.

Keep in mind that for Kerberized remote shell commands, the users UID must be the same on all the nodes. As long as your are using the SP User Management this facility takes care of it.

8.7.6 Never-Expiring Ticket

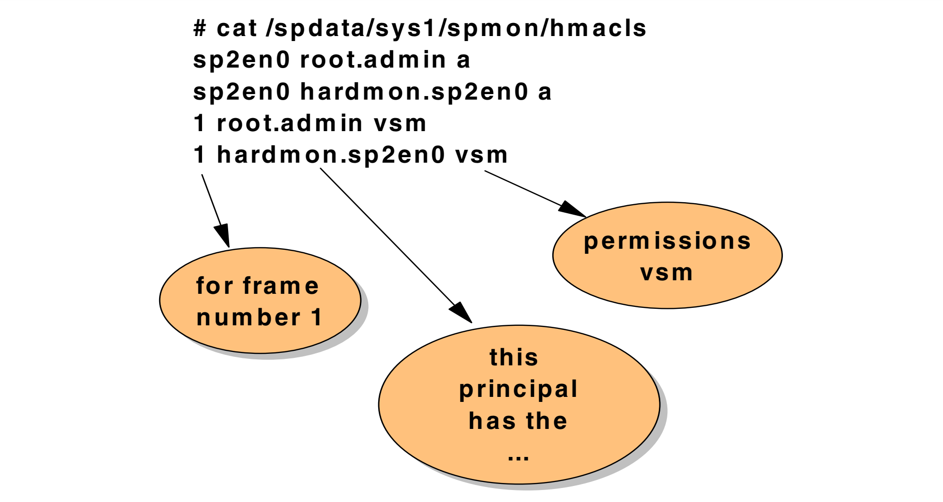
You may wonder why is it possible to get a never-expiring ticket-granting-ticket for the rcmd.<node> principal by executing `rcmdtgt` without issuing any password?

```
sp2n01:/# /usr/lpp/ssp/rcmd/bin/rcmdtgt
sp2n01:/# klist
Ticket file:    /tmp/tkt0
Principal:      rcmd.sp2n01@SP2EN0

  Issued           Expires          Principal
Apr 6 17:25:22  Never           krbtgt.SP2EN0@SP2EN0
```

The authentication is done undercover, but instead of issuing a password the contents of the /etc/krb-srvtab is used. The same processes as described in 8.7.2, “Contents and Purpose of the Ticket-Granting-Ticket (tgt)” on page 169 are done undercover. The authentication server sends a packet back to the client that is encrypted with the client’s private key. The password stored under /etc/krb-srvtab can decrypt it, consequently the ticket-granting-ticket can be obtained.

8.7.7 Hardmon Access Control List - hmacls File

As mentioned in 8.3, “Kerberos Principals” on page 161, hardmon is a Kerberos-protected service that is only available on the CWS. Once you have permission to use this service, it offers the contact to the hardmon daemon that monitors and controls the serial link to the SP frame. By default, only the root.admin principal (or other first user principal) is allowed to obtain a service ticket for the hardmon service because an access control list file exists for the hardmon service and only root.admin is allowed to use this service. This file is called /spdata/sys1/spmon/hmacls. Figure 35 shows the hmacls file.



```
# cat /spdata/sys1/spmon/hmacls
sp2en0 root.admin a
sp2en0 hardmon.sp2en0 a
1 root.admin vsm
1 hardmon.sp2en0 vsm
```

Figure 35. The hmacls File

There are four different sets of permissions, indicated by a single lowercase character:

- m (monitor) - monitor hardware status
- v (virtual front operator pane) - control/change hardware status
- s (serial access) - access to node’s console via serial port (s1term)
- a (administrative) - use hardmon administrative commands

With this permission scheme it is possible to permit a Kerberos principal just the use of starting Perspectives for monitoring, without permitting the issuing of commands over the serial line. For issuing commands, you have to define a Kerberos principal (see 8.8.2, “Make a Kerberos Principal” on page 181), and then add it to the hmacls file. For example, assume anna.sap is a defined

Kerberos principal and we add an entry for her to the /spdata/sys1/spmon/hmacls file; see the following example. Be careful with this file because having even one blank line means hardmon will not serve anything, even if the hardmon subsystem is active.

```
sp2en0:/# vi /spdata/sys1/spmon/hmacls
sp2en0 root.admin a
sp2en0 hardmon.sp2en0 a
1 root.admin vsm
1 hardmon.sp2en0 vsm
1 anna.sap m
```

Then get a ticket-granting-ticket for anna.sap, as follows:

```
# kinit anna.sap
```

Trying `spmon -d` now would return an error message because the hardmon subsystem has to be stopped and restarted so that the hmacls file is read again:

```
# stopsrc -s hardmon
```

```
# startsrc -s hardmon
```

As Kerberos principal anna.sap, let us now try to use `spmon -d`:

```
sp2en0:/# spmon -d | grep thin
 5       5    thin    on   yes  yes     normal   no   LEDs are blank   no
 6       6    thin    on   yes  yes     normal   no   LEDs are blank   no
 7       7    thin    on   yes  yes     normal   no   LEDs are blank   no
 8       8    thin    on   yes  yes     normal   no   LEDs are blank   no
 9       9    thin    on   yes  yes     normal   no   LEDs are blank   no
10      10    thin    on   yes  yes     normal   no   LEDs are blank   no
11      11    thin    on   yes  yes     normal   no   LEDs are blank   no
12      12    thin    on   yes  yes     normal   no   LEDs are blank   no
13      13    thin    on   yes  yes     normal   no   LEDs are blank   no
14      14    thin    on   yes  yes     normal   no   LEDs are blank   no
```

That works, but when we try to open a write terminal on a node, the following message is returned:

```
sp2en0:/# s1term -w 1 7
s1term: 0026-645 The S1 port in frame 1 slot 7 cannot be accessed.
        It either does not exist or you do not have S1 permission.
```

It does not matter which UNIX user has a ticket for the principal anna.sap. Neither root nor any normal UNIX user who is authenticated as anna.sap is allowed with these permissions in the hmacls file to issue commands to hardmon, other than simple monitoring commands.

8.7.8 Kerberos Database Access Control Lists

The Kerberos database located under the /var/kerberos/database on the authentication server consists of two encrypted files, the principal.pag and principal.dir file. Besides these files, there also exist ACL files that are standard ASCII files:

```
sp2en0:/# ls -l /var/kerberos/database
total 78
-rw-r-----   1 root     system        11 Apr 6 16:40 admin_acl.add
-rw-r-----   1 root     system        11 Apr 6 16:40 admin_acl.get
-rw-r-----   1 root     system        11 Apr 6 16:40 admin_acl.mod
-rw-------   1 root     system      4096 Apr 6 16:41 principal.dir
-rw-------   1 root     system         0 Apr 6 16:38 principal.ok
-rw-------   1 root     system     82944 Apr 6 16:41 principal.pag
```

The contents of these files indicate which Kerberos principal is allowed to do the following:

- Add (*admin_acl.add*) information to the database; for example add a principal.
- Get (*admin_acl.get*) information from the database.
- Modify (*admin_acl.mod*) information in the database.

By default, the very first admin principal, usually root.admin, is included in each file. The principals that can get permissions for the database must have the instance admin. To delegate some of the administrative tasks, you can create another admin principal and allow SP monitoring tools by adding him to the hmacls file (see 8.7.7, “Hardmon Access Control List - hmacls File” on page 177), and allow Kerberos administrative tasks by adding him to the appropriate admin_acl file(s).

8.8 Kerberos Commands

With PSSP 2.3 other convenient commands have been added. They are located under /usr/lpp/ssp/kerberos/bin:

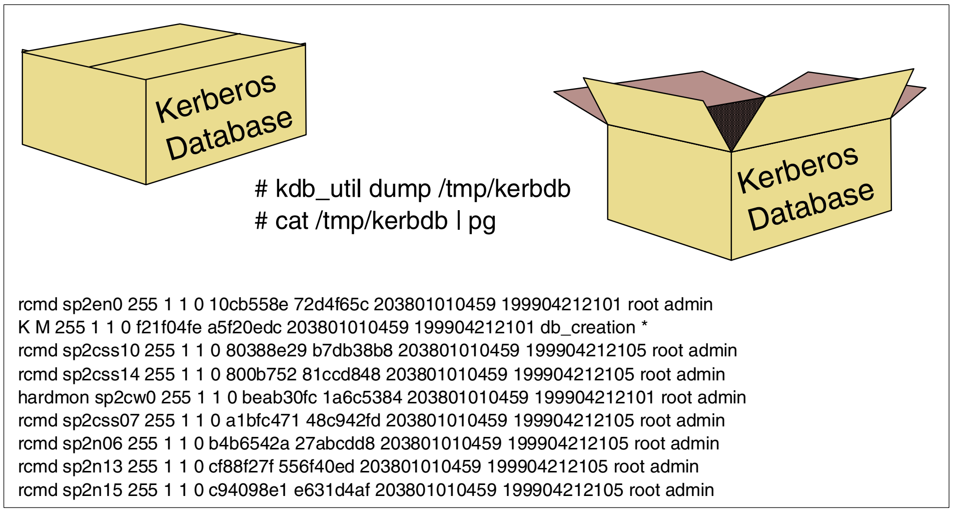
- `lskp` - list Kerberos principal
- `mkkp` - make Kerberos principal
- `chkp` - change Kerberos principal

- `rmkp` - remove Kerberos principal

In the following sections we discuss every command and demonstrate the differences between these new commands and the old Kerberos commands, such as `kdb_util`, `kadmin`, and `kdb_edit`.

8.8.1 List a Kerberos Principal

Before the `lskp` command was provided, two steps were required to list Kerberos principals. Since the Kerberos database is not readable, you first had to dump it into an ASCII file and then look into this file.



```
# kdb_util dump /tmp/kerbdb
# cat /tmp/kerbdb | pg
```

```
rcmd sp2en0 255 1 1 0 10cb558e 72d4f65c 203801010459 199904212101 root admin
K M 255 1 1 0 f21f04fe a5f20edc 203801010459 199904212101 db_creation *
rcmd sp2css10 255 1 1 0 80388e29 b7db38b8 203801010459 199904212105 root admin
rcmd sp2css14 255 1 1 0 800b752 81ccd848 203801010459 199904212105 root admin
hardmon sp2cw0 255 1 1 0 beab30fc 1a6c5384 203801010459 199904212101 root admin
rcmd sp2css07 255 1 1 0 a1bfc471 48c942fd 203801010459 199904212105 root admin
rcmd sp2n06 255 1 1 0 b4b6542a 27abcdd8 203801010459 199904212105 root admin
rcmd sp2n13 255 1 1 0 cf88f27f 556f40ed 203801010459 199904212105 root admin
rcmd sp2n15 255 1 1 0 c94098e1 e631d4af 203801010459 199904212105 root admin
```

Figure 36. Dump the Kerberos Database

This command is useful to find out which principals are currently defined and what the current settings are; for example, the expiration date of a Kerberos account, the ticket lifetime and the key version of this principal, as shown in the following screen:

```
sp2en0:/# lskp
K.M                 tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
changepw.kerberos   tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
default             tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
hardmon.sp2cw0      tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
hardmon.sp2en0      tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
krbtgt.SP2EN0       tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2cw0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2en0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
root.admin          tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
```

8.8.2 Make a Kerberos Principal

When you are going to define a new Kerberos user principal, it has to be included in the Kerberos database and you have to set a password. Let us create the Kerberos principal anna.sap (that no longer exists in the database) with the new `mkkp` command:

```
# mkkp anna.sap
```

This creates a new principal anna with the instance sap in the Kerberos database. But we have not yet set a password for anna.sap. Since these new PSSP Kerberos commands are not interactive, you are not able to set the initial password with the `mkkp` command. But without a password, the principal is blocked. The consequence is that you have to set the password later by getting the `kadmin` prompt and setting the first password with the `change_principal_password` (`cpw`) option. There is always one additional step required, so we recommend that you use the `kadmin` command at once to define a new principal. You will be prompted for the root.admin password first and then set the password for the new principal as in the following example.

Note: Regarding the kadmin command: in case you named your very first Kerberos principal not root.admin but instead decided to chose a different name with instance admin, you have to specify this admin user as a parameter for the `kadmin` command. For instance, if your admin principal is sp.admin, you will get the kadmin prompt by typing:

```
# kadmin -u sp.admin
```

The `kadmin` command, without any options, expects the default root.admin principal. Now let us define anna.sap with the old kadmin command:

```
sp2en0:/# kadmin
Welcome to the Kerberos Administration Program, version 2
Type "help" if you need it.
admin:  ?
Available admin requests:

change_password, cpw     Change a user's password
change_admin_password, cap
                         Change your admin password
add_new_key, ank         Add new user to kerberos database
get_entry, get           Get entry from kerberos database
destroy_tickets, dest    Destroy admin tickets
help                     Request help with this program
list_requests, lr, ?     List available requests.
quit, exit, q            Exit program.
admin:  ank anna.sap
Admin password:
Password for anna.sap:
Verifying, please re-enter Password for anna.sap:
anna.sap added to database.
admin:  quit
Cleaning up and exiting.
```

8.8.3 Change a Kerberos Principal

To change the expiration and ticket lifetime of an existing Kerberos principal, you can use the following command:

```
# chkp [-e <expiration>] [-l <lifetime>] <principal>[.<instance>] ...
```

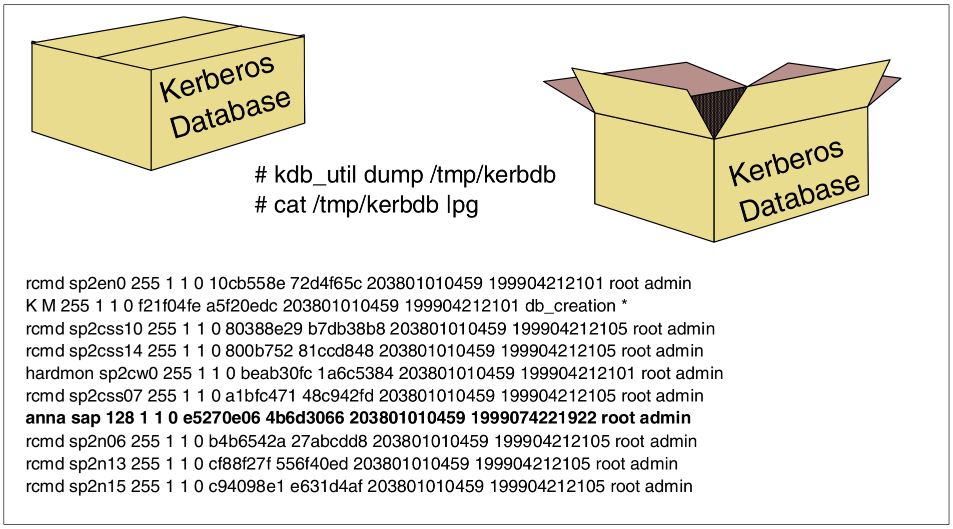
How can you find out when a Kerberos principal account will expire and how long the ticket lifetime for this principal currently is? The best way is to use both commands, `lskp` and `kdb_util dump`, especially when you want to change something. The `lskp` command returns the expiration date and the ticket lifetime in hours and minutes.

However, when you want to change the ticket lifetime of a principal, you cannot specify hours and minutes; instead you need to specify values from 1 to 255. Values 1 to 128 represent multiples of 5 minutes, while lifetime values from 129 to 255 represent intervals from 11 hours and 40 minutes to 30 days (see *IBM Parallel System Support Programs for AIX: Administration Guide,* GC23-3897).

The current value is readable in the dump file of the database. Let us first have a look to the current settings of the principal anna.sap, and then use the `chkp` command to change them.

```
sp2en0:/# lskp |grep anna.sap
anna.sap           tkt-life:    10:40 key-vers: 1  expires: 2037-12-31 23:59
sp2en0:/# chkp -e 2000-04-26 anna.sap
sp2en0:/# lskp |grep anna.sap
anna.sap           tkt-life:    10:40 key-vers: 1  expires: 2000-04-26 23:59
```

As you can see, the expiration date has been changed and the ticket lifetime is 10 hours and 40 minutes. You can look up this value by using the `kdb_util dump` command.



```
# kdb_util dump /tmp/kerbdb
# cat /tmp/kerbdb lpg
```

```
rcmd sp2en0 255 1 1 0 10cb558e 72d4f65c 203801010459 199904212101 root admin
K M 255 1 1 0 f21f04fe a5f20edc 203801010459 199904212101 db_creation *
rcmd sp2css10 255 1 1 0 80388e29 b7db38b8 203801010459 199904212105 root admin
rcmd sp2css14 255 1 1 0 800b752 81ccd848 203801010459 199904212105 root admin
hardmon sp2cw0 255 1 1 0 beab30fc 1a6c5384 203801010459 199904212101 root admin
rcmd sp2css07 255 1 1 0 a1bfc471 48c942fd 203801010459 199904212105 root admin
anna sap 128 1 1 0 e5270e06 4b6d3066 203801010459 1999074221922 root admin
rcmd sp2n06 255 1 1 0 b4b6542a 27abcdd8 203801010459 199904212105 root admin
rcmd sp2n13 255 1 1 0 cf88f27f 556f40ed 203801010459 199904212105 root admin
rcmd sp2n15 255 1 1 0 c94098e1 e631d4af 203801010459 199904212105 root admin
```

Figure 37. Kerberos Database Dump File

The current ticket lifetime of anna.kerb is 128, or 10 hours and 40 minutes. So let us change this value to 15 minutes.

```
sp2en0:/# chkp -l 3 anna.sap
sp2en0:/# lskp | pg
K.M                tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
anna.sap           tkt-life:    00:15 key-vers: 1  expires: 2000-04-26 23:59
changepw.kerberos  tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
default            tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
hardmon.sp2cw0     tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
hardmon.sp2en0     tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
krbtgt.SP2EN0      tkt-life: 30d       key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css01      tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp2css05      tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
^C
```

With the database dump file, you get the ticket lifetime value 3:

```
sp2en0:/# kdb_util dump /temp/kerbdb
sp2en0:/# cat /tmp/kerbdb | grep anna
anna sap 3 1 1 0 e5270e06 4b6d3066 200004270459 199904162132 root admin
```

As mentioned before, these new PSSP Kerberos commands make handling Kerberos more convenient, but you can also use the old commands. For changing Kerberos principals, the corresponding command is `kdb_edit`:

```
p2en0:/# kdb_edit
Opening database...

Enter Kerberos master key:

Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: anna
Instance: sap

Principal: anna, Instance: sap, kdc_key_ver: 1
Change password [n] ? n
Expiration date (enter yyyy-mm-dd) [ 2000-04-27 ] ? 1999-04-16
Max ticket lifetime [ 3 ] ? 10
Attributes [ 0 ] ?
Edit O.K.
Principal name: <just press Enter to exit> # comment by author
#
```

8.8.4 Remove a Kerberos Principal

Until PSSP 2.2, deleting a Kerberos principal consisted of three steps: dump the Kerberos database to a file as shown in the examples; delete the line of the Kerberos principal you want to remove; load the database again with the `kdb_util load <file>` command. With the `rmkp` command, however, deleting a principal became very easy. To delete anna.sap, just type:

```
# rmkp anna.sap
```

Then confirm the removal.

For all new Kerberos commands introduced with PSSP 2.3, only root authority is required; no ticket-granting-ticket is needed.

8.9 Reinitializing Kerberos without Rebooting the Nodes

In a situation where you are forced to fix Kerberos problems but you cannot afford to reboot the nodes in customize mode, or effect the running switch by executing the `pssp_script` manually, all the Kerberos-related files can be generated and distributed without any reboot. Here is a short example of how you can set up your Kerberos environment from scratch by using parts of these procedures.

8.9.1 Removing or Reinitializing the Kerberos Subsystem

Since PSSP 2.3, reinitializing the Kerberos subsystem is quite easy. You just issue `setup_authent` and confirm the replacement of the Kerberos setup. The database is replaced, the Kerberos subsystems are stopped and restarted, and if you already have an installed SP, all nodes will be set automatically to *customize,* and `setup_server` will be started to rebuild the new krb-srvtab files for the nodes. What is left to do is the distribution of the appropriate files to the nodes.

If you want to keep some of the Kerberos configuration files, then follow the steps as in older PSSP versions before a rerun of `setup_authent` takes place.

Until PSSP 2.3 you had to delete the Kerberos files manually. Since it is possible to keep some of the files, especially if you edited them, we will not only describe how to delete all Kerberos files, but also tell you which files can be kept.

```
# rm /.k /.klogin
```

The master key /.k must be removed, but /.klogin can be kept.

```
# rm /etc/krb*
```

This command deletes /etc/krb.conf, /etc/krb.realms and /etc/krb-srvtab. While krb-srvtab has to be removed, krb.conf and krb.realms files can stay. `Setup_authent` will use the contents for the new setup.

```
# kdb_destroy
```

This command deletes the database files, meaning the principal.pag and the principal.dir files under /var/kerberos/database. The ACL files for the Kerberos database remain. In each ACL file there is an entry for the root.admin principal; consequently, if you set up your Kerberos with a different first admin principal than root.admin, you will run into trouble.

Therefore, to delete all these files use the command:

```
# rm /var/kerberos/database/*
```

As soon as you have deleted the Kerberos database and the master key /.k, the `setup_authent` command will not run into the replacement loop but instead set up Kerberos again based on the existing files. Now you can rerun:

```
# setup_authent
```

Up to PSSP 2.3, the krb-srvtab files are recreated automatically.

Since we plan neither a reboot of the nodes nor starting the `pssp_script` manually, we can set all the nodes back to *response=disk* and then distribute the files by file transfer. For the bootp_response value customize, no NIM resource allocation has taken place; therefore, it is sufficient to change this response value in the SDR without running `setup_server`.

```
# spbootins -r disk -l 1,5,6,7 -s no
```

All new-srvtab files for the nodes are stored under the /tftpboot directory on the CWS. At a minimum, these files have to be copied to the nodes. If nothing changed in your configuration, meaning you are using the same realm name and the same adapter names, the /etc/krb.conf and /.klogin files on the nodes can still be used. Regardless of whether the Kerberos client could use the old files or really needs the new ones, we will demonstrate the transfer of all files that are required on a Kerberos client for sp2n01. Due to the long outputs of the ftp command, we cut the messages out and show only the essential part:

```
sp2en0:/# ftp sp2n01
ftp> bin
ftp> put /tftpboot/sp2n01-new-srvtab /etc/krb-srvtab
ftp> put /etc/krb.conf
ftp> put /etc/krb.realms
ftp> put /.klogin
ftp > quit
```

This procedure has to be repeated for each node. After the distribution of the new-srvtab files to the nodes, delete them under /tftpboot on the CWS, since they are no longer needed in this place and this directory permits access for tftp.

```
# rm /trftpboot/*new*
```

8.10 Recreating the krb-srvtab File for a Node

The krb-srvtab files were always created on the primary authentication server (usually the CWS), and then distributed to the appropriate nodes. You have two possibilities to build a new krb-srvtab file for a node.

- By using the `create_krb_files` (or `setup_server`) command
- By using the `ext_srvtab` command

8.10.1 The create_krb_files Wrapper

Since `create_krb_files` only creates the krb-srvtab file for nodes that have a bootp_response value of customize, install or migrate in the SDR, you have to set the node to customize before running the following command:

```
# spbootins -r customize -l 6 -s no
```

This command sets node 6 to customize and `setup_server` will not run due to the -s option. We only need the wrapper `create_krb_files` (refer to Figure 31 on page 167). This command creates the new-srvtab file for node 6 under /tftpboot.

When `create_krb_files` has finished and you have obtained all needed <node>-new-srvtab files, do not forget to set the appropriate node(s) back to disk:

```
# spbootins -r disk -l 6 -s no
```

In this case, it is sufficient just to set the bootp_response value in the SDR back to disk. For “customize”, no resource has been allocated, therefore it is not necessary to run the `unallnimres` command (or `setup_server`).

Now you can transfer the /tftpboot/<node>-new-srvtab to the concerned node(s) under /etc/krb-srvtab.

8.10.2 The ext_srvtab Wrapper

The `ext_srvtab` command works independently of the SDR; therefore, it is not necessary to change any value in the SDR before creating new-srvtabs. This command extracts the srvtab directly from the Kerberos database. This works only per adapter (in other words, per rcmd principal). Following is the syntax of this command:

```
# ext_srvtab [-n] [-r realm] [instance ...]
```

If you use the -n option you will not be prompted for the Kerberos master key as long as a /.k file exists. Usually the -r option is not needed either, since the realm name of the rcmd principal is the same as your local realm in which you are authenticated as root.admin. The instance of a service principal is always the adaptername (short form, without domain) over which the service is offered. Now let us extract the srvtab for node 7 that has one Ethernet and one switch adapter. Due to the equality of adapter names and instance names of the rcmd principals, the command must look as follows:

```
# ext_srvtab -n sp2n07 sp2css07
```

The result is two new-srvtab files in the current directory. What is still left to do is the concatenation of all the srvtab files belonging to a node into one srvtab file (whose name is free-choosable), and then the transfer to the corresponding node as /etc/krb-srvtab file.

```
# ext_srvtab -n sp2n07 sp2css07
Generating 'sp2n07-new-srvtab'....
Generating 'sp2css07-new-srvtab'....
# ls -la | grep 07
-rw-------   1 root     system        30 Mar 24 14:23 sp2css07-new-srvtab
-rw-------   1 root     system        28 Mar 24 14:23 sp2n07-new-srvtab
# cat sp2n07-new-srvtab sp2css07-new-srvtab > node07-new-srvtab
# ls -la | grep srvtab
-rw-r--r--   1 root     system        58 Mar 24 14:25 node07-new-srvtab
-rw-------   1 root     system        30 Mar 24 14:23 sp2css07-new-srvtab
-rw-------   1 root     system        28 Mar 24 14:23 sp2n07-new-srvtab
# ftp sp2n07
Connected to sp2n07.
220 sp2n07 FTP server (Version 4.1 Tue Mar 17 14:00:13 CST 1998) ready.
Name (sp2n07:root):
331 Password required for root.
Password:
230 User root logged in.
ftp> bin
200 Type set to I.
ftp> put node07-new-srvtab /etc/krb-srvtab
200 PORT command successful.
150 Opening data connection for /etc/krb-srvtab.
226 Transfer complete.
58 bytes sent in 0.02141 seconds (2.645 Kbytes/s)
local: node07-new-srvtab remote: /etc/krb-srvtab
ftp> quit
221 Goodbye.
# dsh -w sp2n07 date
sp2n07: Fri Apr 16 15:08:59 EDT 1999
```

Do not forget to delete the <node>new-srvtab file under /tftpboot as soon as you have transferred them to the appropriate node. This is always done when the customize script (`pssp_script`) is running on a node. After transferring the Kerberos-related files, the /tftpboot/<node>-new-srvtab is deleted. It is recommended that you do not keep these srvtab files hanging around under a directory that permits tftp.

8.11 Merging Two (or more) SP Systems to One Kerberos Realm

One advantage of having an RS/6000 SP is the centralized administration of the whole system: you can distribute, install, to check up something in parallel. One command works for all your nodes. Though some customers have several SP systems in different locations due to safety prerequisites or other reasons, these systems often are administered by the same people. In such configurations it is desirable to work in a common secure environment including all the SPs. Otherwise, each time you change from one SP to another for administrating the system, you have to change authenticate to a different Kerberos realm.

It is preferable to have all SPs belong to a single realm. Here is the solution for having one Kerberos realm spread over two (or even more) SP systems.

Assume we have two SP systems, sp4en0 and sp5en0, and sp5en0 will be integrated to the realm of sp4en0. That means the CWS sp4en0 will become the authentication server for both SP systems. Later, we will set up sp5en0 as a secondary authentication server for this realm. Figure 38 on page 190 illustrates this configuration.

The common realm name will be SP4EN0, but as you know, you are free to choose the realm name and create your own /etc/krb.conf file before issuing `setup_authent` (see 8.2.2, “The Kerberos Configuration File” on page 160 for further information).

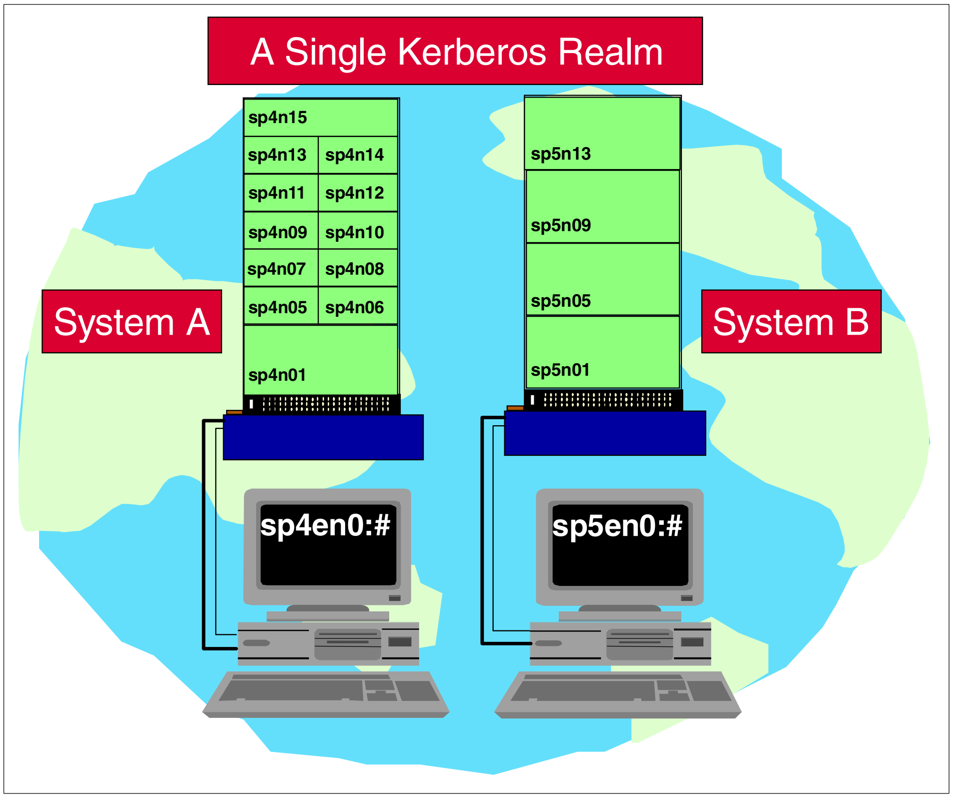


Figure 38. Two SP Systems in a Single Kerberos Realm

8.11.1 Deleting an Authentication Server Setup

When you are planning to spread one Kerberos realm over two SP systems, the starting point usually is that you already have two installed systems with an authentication server for each system (meaning for each realm). The integration of SP system B into the realm of SP system A makes the deletion of the authentication server on sp5en0 (the CWS of SP system B) necessary. In the first step, sp5en0 will become a normal Kerberos client just like all the other nodes under the control of the authentication server on sp4en0.

You may wonder, which files does the Kerberos master use, and which subsystems belong to the Kerberos master? Besides the Kerberos configuration files that are stored on the server and the clients, the server is owner of the Kerberos database. So let us delete the related files and remove the subsystems on sp5en0:

```
sp5en0:/# rm /var/kerberos/database/*
sp5en0:/# rm /.k
sp5en0:/# rm /etc/krb-srvtab
sp5en0:/# rm /etc/krb.conf
sp5en0:/# stopsrc -s kadmind
0513-044 The stop of the kadmind Subsystem was completed successfully.
sp5en0:/# stopsrc -s kerberos
0513-004 The Subsystem or Group, kerberos, is currently inoperative.
sp5en0:/# rmssys -s kadmind
0513-083 Subsystem has been Deleted.
sp5en0:/# rmssys -s kerberos
0513-083 Subsystem has been Deleted.
```

After these commands are run, sp5en0 is no longer an authentication server. We intentionally did not delete the /etc/krb.realms and /.klogin files since we can use these files later to create the global /etc/krb.realms and /.klogin files on the new authentication server. Since the /etc/krb-srvtab and /etc/krb.conf files will be overwritten, it is not necessary to delete them, but the subsystems and the database will have to be deleted.

8.11.2 Provide Name Resolution and Time Synchronization

Many of the problems with Kerberos are based on incorrect name resolution, so you should insure that the name resolution for the nodes in SP system A, as well for the nodes and CWS of SP system B, is correct. It does not matter whether you are using the /etc/hosts or domain name service. For the single realm including two SP systems, we recommend that you to add all the hosts of the SP system B to the /etc/hosts on the CWS of SP system A (in our example, sp4en0).

For more information about Kerberos and IP configuration, refer to *IBM Parallel System Support Programs for AIX: Administration Guide,* GC23-3897.

Since Kerberos packets have a lifetime of 5 minutes, the time difference between the hosts must not exceed this limit. Therefore, when you plan to spread one Kerberos realm across SP systems, you have to insure that the time difference between these systems is limited. Otherwise, your Kerberos authentication and communication will not work. One solution for solving this problem is to use an external time server to synchronize the time services within the SPs.

8.11.3 Adjust the /etc/krb.realms File

The CWS of SP system A (sp4en0) will become the authentication server for the whole realm; that means the Kerberos database will be located on this host. In addition, all Kerberos clients will get the Kerberos-related files from this server, therefore we will adjust all the files on the future authentication server (sp4en0).

The /etc/krb.realms file maps adapter names to an authentication realm. An /etc/krb.realms file with all the adapter names for SP system B already exists, but for our purpose, it is pointing to the wrong realm. Consequently, you must copy the krb.realms file of sp5en0 to another file on sp4en0, update the realm entry for every adapter to the new realm name, and then concatenate this file with the krb.realms file of sp4en0 (in our case, SP4EN0). Finally, this file should to look like this:

```
sp4en0:/# cat /etc/krb.realms
sp4cw0 SP4EN0
sp4cw0 SP4EN0
sp4n01 SP4EN0
sp4css01 SP4EN0
sp5n01 SP4EN0
sp5n01x SP4EN0
sp4n05 SP4EN0
sp4css05 SP4EN0
sp4n06 SP4EN0
sp4css06 SP4EN0
sp4n07 SP4EN0
sp4css07 SP4EN0
sp4n08 SP4EN0
sp4css08 SP4EN0
sp4n09 SP4EN0
sp4css09 SP4EN0
sp5n09 SP4EN0
sp5n09x SP4EN0
^C
```

8.11.4 Extend the /.klogin File

Imagine are you working on sp5n09 on SP system B, you own a ticket-granting-ticket for root.admin, and you want to set up a dsh command to a node belonging to SP system A, as follows:

```
sp5n09:/# dsh -w sp4n07 date
```

After the authentication processes (tgt, service ticket) complete, it will be verified whether node sp5n09 is allowed to login to sp4n07 to issue the command. This will be checked in the /.klogin file; therefore, all principals of both SPs have to have an entry in this file. We recommend that you include not only the rcmd.<en0_name> principals based on the internal Ethernet, but also the rcmd.<css0_name> principals so that you are allowed to set up Kerberized remote shell commands over the switch.

You can create the global /.klogin file in the same way as you created the /etc/krb.realms file. However, you have to delete the root.admin principal of the old realm; it must no longer exist. We have one root.admin for our global realm, that is all. The /.klogin will look like this, and the order does not matter:

```
p4en0:/# cat /.klogin | pg
root.admin@SP4EN0
rcmd.sp4en0@SP4EN0
rcmd.sp4cw0@SP4EN0
rcmd.sp5en0@SP4EN0
rcmd.sp5cw0@SP4EN0
rcmd.sp4n01@SP4EN0
rcmd.sp4css01@SP4EN0
rcmd.sp5n01@SP4EN0
rcmd.sp5n01x@SP4EN0
rcmd.sp4n05@SP4EN0
rcmd.sp4css05@SP4EN0
rcmd.sp5n05@SP4EN0
rcmd.sp5n05x@SP4EN0
^C
```

8.11.5 Adding Principals for Remote Nodes

There are two ways to add the rcmd service principal for each adapter of SP system B to the Kerberos database on sp4en0:

- Create rcmd principals with the `kadmin` command.
- Use a file as input for the `add_principal` command.

When you use the `kadmin` prompt, you have to define one rcmd principal after the other and set a password for each of them. It is required to have a valid password for every principal; otherwise, you will not be allowed to log on or authenticate as this principal. On the other hand, you are never prompted for the rcmd principals' password because the login routine for the rcmd principal is done under cover. The `rcmdtgt` command asks for a ticket-granting-ticket (tgt). Then the authentication server sends a packet back containing the tgt encrypted with the private key of this node (see also 8.7, “Tickets and Keys” on page 168). The /etc/krb-srvtab on the node contains the password for this principal. Finally, with this srvtab file, the packet can be decrypted locally so that the tgt can be obtained.

Because of this mechanism, you do not need to keep these passwords in mind. The second way to define all these principals to the Kerberos database and provide them with a password can be realized by creating a file that contains all principals and passwords and by issuing the `add_principal` command that uses this file as input.

8.11.5.1 Creating rcmd Principals with the kadmin Command

The best way to do this is to have a list with all the adapter names that should be integrated. As example, we will define the adapters of sp5n01 (the first node of SP system B) with the `kadmin` command:

```
sp4en0:/# kadmin
Welcome to the Kerberos Administration Program, version 2
Type "help" if you need it.
admin:  ank rcmd.sp5n01
Admin password:
Password for rcmd.sp5n01:
Verifying, please re-enter Password for rcmd.sp5n01:
rcmd.sp5n01 added to database.
admin:  ank rcmd.sp5n01x
Admin password:
Password for rcmd.sp5n01x:
Verifying, please re-enter Password for rcmd.sp5n01x:
rcmd.sp5n01x added to database.
admin:  quit
Cleaning up and exiting.
```

You could define every rcmd principal within one kadmin session. You have to set a password and verify it, but you can forget it afterwards. Now let us check if these principals are defined:

```
sp4en0:/# lskp | grep sp5
rcmd.sp5n01         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp5n01x        tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
```

8.11.5.2 Use the add_principal Command

The advantage of creating a file with all principals that you are going to define is that you run the `add_principal` command just once. The disadvantage of this method is that you have to also provide the passwords in this file.

First of all we create a file that has read and write permission only for the owner, say /tmp/addnew, in order to insure that no other person can look into this file during the procedure of creating principals.

```
# (umask 066 ; vi /tmp/addnew)
```

This command creates a file /tmp/addnew with permission -rw-------. The brackets are needed to avoid setting your default umask to this value. It is only valid for the following command. You can also touch the file with your default umask and then change the permissions with `chmod 600 /tmp/addnew`.

Now an entry for each principal that should be defined is needed, followed by a blank and then the password of your choice. Let us define the rcmd principals for node 5 and the CWS of SP system B. The file structure has must look like this:

```
sp4en0:/# cat /tmp/addnew
rcmd.sp5en0 siw96ed62h
rcmd.sp5cw0 926fkduen3
rcmd.sp5n05 93jd7dj49f
rcmd.sp5n05x uejf7390id
```

The `add_principal` command, located under /usr/lpp/ssp/kerberos/bin, takes the first field as the name of the principal that should be defined and the second field is used as password. We issue the `add_principal` command, then have a look into the database to see which principals are already defined:

```
sp4en0:/# add_principal -v /tmp/addnew
add_principal: rcmd.sp5en0 added to database.
add_principal: rcmd.sp5cw0 added to database.
add_principal: rcmd.sp5n05 added to database.
add_principal: rcmd.sp5n05x added to database.
sp2en0:/# lskp | grep sp5
rcmd.sp5cw0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp5en0         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp5n01         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp5n01x        tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp5n05         tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
rcmd.sp5n05x        tkt-life: Unlimited key-vers: 1  expires: 2037-12-31 23:59
```

We do not delete the /tmp/addnew file immediately because we can also use it to create the new-srvtab files for the new principals; it saves time to use this file as input.

8.11.6 Creating the srvtab Files for the Remote Nodes

As soon as the principals for all adapters names are defined in the Kerberos database, the krb-srvtab files for the appropriate nodes can be extracted. Since the newly created principals (in other words, the appropriate adapter name of these principals) are not part of the SDR on our CWS sp4en0, we cannot use the `create_krb_files` wrapper to create the krb-srvtab files (see Figure 31 on page 167). We are forced to use the `ext_srvtab` command, which is SDR-independent and works directly with the Kerberos database entries.

Because all newly created principals are listed in the /tmp/addnew file (except the principals for node 1), we can use this file as input for the `ext_srvtab` command. The syntax is:

```
# ext_srvtab [-n] [-r realm] instance [instance ...]
```

The instance names are equal to the adapter names. The entries in /tmp/addnew look like rcmd.sp5n05 <password>, so we will cut the instance names out and use them as input for `ext_srvtab`. The new-srvtabs are created in the current directory. To make work easier, it is recommended to create new directory /tftpboot/srv and change to it. Now all new-srvtab files will be generated in this subdirectory.

```
sp4en0:/# cd /tftpboot/srv
sp4en0:/tftpboot/srv# ext_srvtab -n `cat /tmp/addnew | cut -d’.’ \
>-f2 | awk '{print $1}'`
Generating 'sp5en0-new-srvtab'....
Generating 'sp5cw0-new-srvtab'....
Generating 'sp5n05-new-srvtab'....
Generating 'sp5n05x-new-srvtab'....
```

Do not forget to remove the /tmp/addnew file with the passwords for the rcmd principals if you have used the `add_principal` command.

```
# rm /tmp/addnew
```

Since the rcmd principals for node 1 have been defined with the `kadmin` command, they were not included in our /tmp/addnew list. Let us now create these srvtabs by naming the principals as parameters and then have a look at the directory:

```
sp4en0:/tftpboot/srv# ext_srvtab -n sp5n01 sp5n01x
Generating 'sp5n01-new-srvtab'....
Generating 'sp5n01x-new-srvtab'....
sp2en0:/tftpboot/srv# ls -l
total 48
-rw-------   1 root     system        28 Mar 31 19:14 sp5cw0-new-srvtab
-rw-------   1 root     system        28 Mar 31 19:14 sp5en0-new-srvtab
-rw-------   1 root     system        28 Mar 31 19:30 sp5n01-new-srvtab
-rw-------   1 root     system        29 Mar 31 19:30 sp5n01x-new-srvtab
-rw-------   1 root     system        28 Mar 31 19:14 sp5n05-new-srvtab
-rw-------   1 root     system        29 Mar 31 19:14 sp5n05x-new-srvtab
```

The result is one new-srvtab file for each adapter. Therefore, we have to concatenate the srvtab files belonging to one node before we distribute them.

```
# cat sp5n01-new-srvtab sp5n01x-new-srvtab > sp5n01-new-srv
```

Follow this procedure for every node.

8.11.7 Distributing the Kerberos Files to the Remote Nodes

The last step consists of distributing new-srvtab files to the nodes as well to the second CWS (which is no longer an authentication server). Figure 28 on page 158 shows an example for node 1.

The files that a Kerberos client needs are, at minimum:

- /etc/krb.conf
- /etc/krb.realms
- /etc/krb-srvtab
- /.klogin

All files except the /etc/krb-srvtab files are copies of the Kerberos server files.

```
sp4en0:/tmp# ftp sp5n01
Connected to sp5n01.
220 sp5n01 FTP server (Version 4.1 Tue Mar 17 14:00:13 CST 1998) ready.
Name (sp5n01:root):
331 Password required for root.
Password:
230 User root logged in.
ftp> bin
200 Type set to I.
ftp> put /tftpboot/srv/sp5n01-new-srv /etc/krb-srvtab
200 PORT command successful.
150 Opening data connection for /etc/krb-srvtab.
226 Transfer complete.
29 bytes sent in 0.02214 seconds (1.279 Kbytes/s)
local: /tftpboot/srv/sp5n01-new-srv remote: /etc/krb-srvtab
ftp> put /etc/krb.conf
200 PORT command successful.
150 Opening data connection for /etc/krb.conf.
226 Transfer complete.
36 bytes sent in 0.000413 seconds (85.12 Kbytes/s)
local: /etc/krb.conf remote: /etc/krb.conf
ftp> #### and also /etc/krb.realms and /.klogin ####
```

Once the distribution of the Kerberos-related files is finished, your Kerberos environment with two SP systems in one Kerberos realm is ready to use.

8.12 Setting Up a Secondary Authentication Server

Due to the dependency of the authentication server for getting tickets, especially if you are serving several SP systems from one authentication server, we set up a secondary authentication server. Every workstation, even if it is not part of the realm, can become a secondary authentication server. In our example it makes sense to define the second CWS, that became a normal Kerberos client during the fusion to one singe Kerberos realm, as a secondary server.

The `setup_authent` script that will run to define the secondary server does not allow you to define an SP node as a secondary server. As soon as it realizes that you are working on an SP node, the script exits. This works as designed because usually the nodes are the login hosts for your user and it is not recommended to place a copy of the database on a “user” machine. Nevertheless, a workaround for that is described in Appendix C, “Secondary Authentication Server on an SP Node” on page 255.

Our starting point consists of a Kerberos realm spread across two SP systems. The Kerberos configuration after the setup of a secondary authentication server will look like Figure 39.

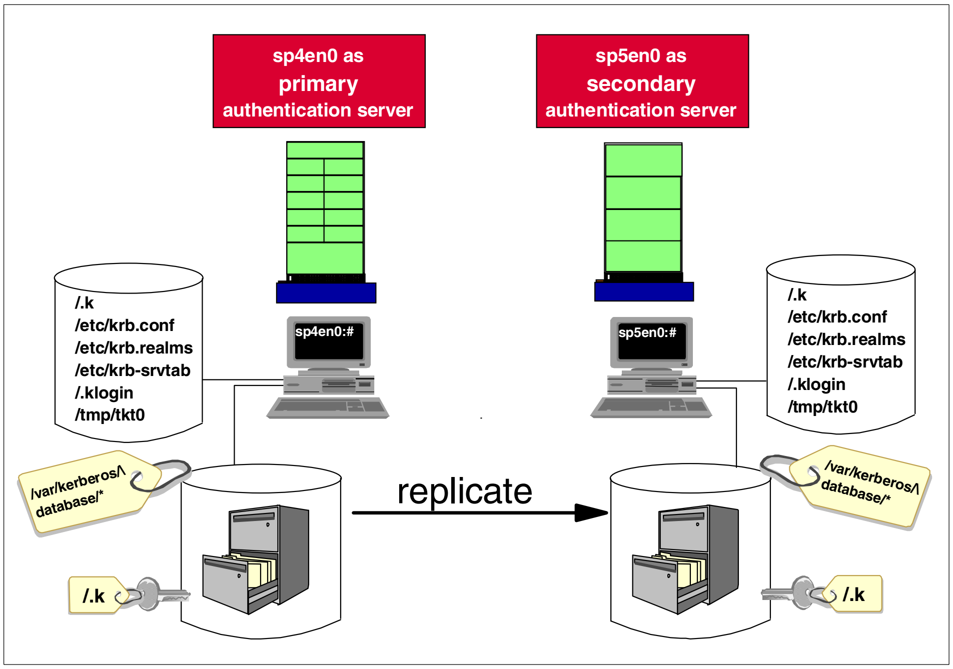


Figure 39. Secondary Authentication Server

Due to an additional entry in the /etc/krb.conf file, sp5en0 will be identified as a secondary authentication server. The secondary server gets a read-only replica of the Kerberos database from the primary authentication server. That means the secondary server is able to do Kerberos authentication. The secondary server can generate tickets, but any configuration changes should only be applied to the primary authentication server. Every node within this realm has the same /etc/krb.conf file pointing to sp5en0 as the secondary server. If the communication with the primary server fails, the clients ask the secondary server for authentication.

Assuming that the future secondary server is already a client of the Kerberos realm, the following steps are necessary to set up the secondary server:

- Install the Kerberos filesets on the future secondary server (if necessary).
- Edit the /etc/krb.conf file and distribute it to all hosts belonging to the realm.
- Run `setup_authent` on the secondary authentication server.
- Add a /usr/kerberos/etc/push-krop entry to the crontab on the primary server.

8.12.1 Install Kerberos-Related Filesets on a Secondary Server

The authentication server and client part of the PSSP software is needed on the secondary server, as follows:

- ssp.authent
- ssp.clients

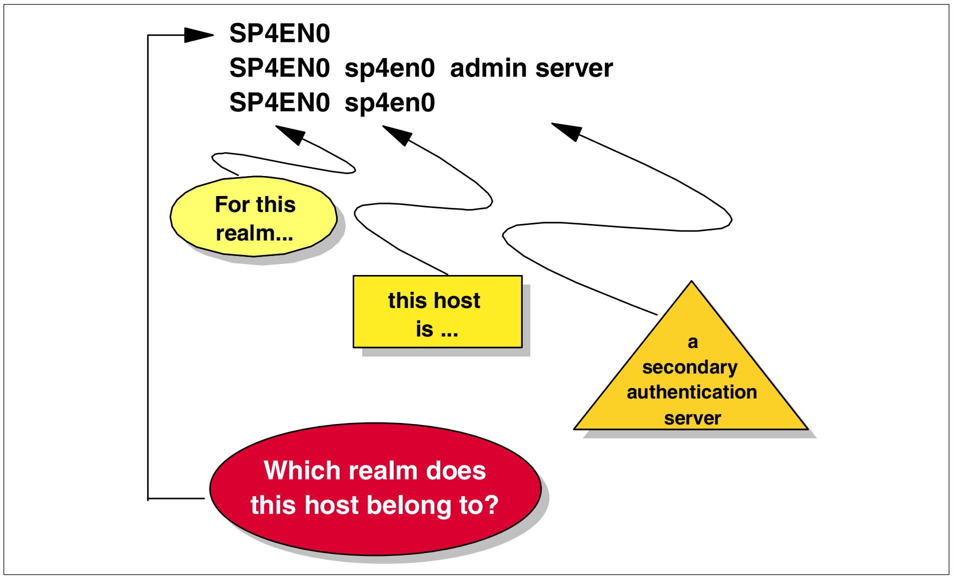
In our example, the sp5en0 had been a Kerberos authentication server before, consequently the filesets are already installed.

Insure that the name resolution for all hosts belonging to the Kerberos realm works properly.

8.12.2 /etc/krb.conf File Modification and Distribution

Since we have a working Kerberos environment, it makes sense to edit the /etc/krb.conf file on the primary authentication server and then distribute it with the parallel copy command to all nodes.

The /etc/krb.conf file contents are explained in 8.2.2, “The Kerberos Configuration File” on page 160. The second line points to the *admin server*, meaning to the primary authentication server. To define a secondary authentication server an additional entry is required:



```
SP4EN0
SP4EN0  sp4en0  admin server
SP4EN0  sp4en0
```

For this realm...
this host is ...
a secondary authentication server
Which realm does this host belong to?

Figure 40. The /etc/krb.conf File with Secondary Authentication Server

Only one admin server entry (primary authentication server) should be defined for a realm, but you can define several secondary servers. The definitions for the secondary authentication server look like the entry for sp5en0 in Figure 39 on page 198.

To distribute this file to all the nodes of our realm we use the parallel copy command `pcp`. Details on the parallel copy command and its possible options are covered in 8.13, “Working with the WCOLL Variable” on page 202.

Assume the file /tmp/allhosts contains all nodes of both SP systems, plus the second CWS. First we will set the WCOLL variable to that file, and then copy the /etc/krb.conf to all nodes in parallel:

```
sp2en0:# export WCOLL=/tmp/allhosts
sp2en0:# pcp '' /etc/krb.conf
```

8.12.3 Run setup_authent on the Secondary Authentication Server

As mentioned in 8.5, “setup_authent, setup_server and the Kerberos Database” on page 164, the `setup_authent` script first checks if a /etc/krb.conf file already exists. In our case, when we execute `setup_authent` on our future secondary server (sp5en0), the script detects the entry for sp5en0 in the krb.conf file and starts the appropriate part of `setup_authent` to make this host a secondary server. See the output of our example:

```
sp5en0:/# setup_authent
*********************************************************************
                               ATTENTION

 You are attempting to redo the authentication setup on your Control
 Workstation.  Since you have not yet executed install_cw, there are
 no special considerations or risks involved.

 Do you want to replace your existing setup?
 *********************************************************************

Enter y or n: y
 ************************************************************************
                Logging into Kerberos as an admin user

[...]

 For more information, see the kinit man page.
 ************************************************************************
setup_authent: Enter name of admin user: root.admin
Kerberos Initialization for "root.admin"
Password:
add_principal: 2502-037 rcmd.sp5cw0 already exists in database.
add_principal: 2502-037 hardmon.sp5cw0 already exists in database.
add_principal: 2502-037 rcmd.sp5en0 already exists in database.
add_principal: 2502-037 hardmon.sp5en0 already exists in database.
0513-085 The kpropd Subsystem is not on file.
0513-084 There were no records that matched your request.
sp5cw0: success.
sp5cw0:         Succeeded
0513-071 The kpropd Subsystem has been added.
0513-059 The kpropd Subsystem has been started. Subsystem PID is 21896.
0513-085 The kerberos Subsystem is not on file.
0513-084 There were no records that matched your request.
0513-071 The kerberos Subsystem has been added.
0513-059 The kerberos Subsystem has been started. Subsystem PID is 20772.
```

The kpropd and Kerberos subsystems have been added and we already received a read-only replica of the Kerberos database. Due to the read-only functionality on a secondary server, the kadmind subsystem is not installed. All changes to the database must be, and can only be, done on the primary authentication server.

The kpropd is the daemon that receives the copy of the Kerberos database from the primary authentication server. Now let us have look to the contents of the /var/kerberos/database directory on the secondary server, because there is a difference from the primary server:

```
sp5en0:/# cd /var/kerberos/database
sp5en0:/var/kerberos/database# ls -l
total 83
-rw-r-----   1 root     system         0 Jul 20 17:51 admin_acl.add
-rw-r-----   1 root     system        11 Jul 20 17:51 admin_acl.get
-rw-r-----   1 root     system         0 Jul 20 17:51 admin_acl.mod
-rw-------   1 root     system      4096 Jul 20 17:51 principal.dir
-rw-------   1 root     system         0 Jul 20 17:51 principal.ok
-rw-------   1 root     system     82944 Jul 20 17:51 principal.pag
```

Since the secondary authentication server gets a read-only replica of the Kerberos database, the admin_acl.add and admin_acl.mod files under /var/kerberos/database have no entries. The only entry for root.admin is in the admin_acl.get file, which indicated which Kerberos principal is allowed to get information from the Kerberos database.

8.13 Working with the WCOLL Variable

When your realm consists of several SP systems or includes external workstations (see 9.1, “Integration of External Workstations to the SP Kerberos Realm” on page 205), it is particularly necessary to work with the WCOLL variable since the usual hostlist arguments as parameters for Kerberized remote shell commands get their information from the SDR. The best way to do this is to create a file containing all hosts of the realm, for example /tmp/allhosts. Each host entry must begin on a new line:

```
p2en0:/# cat /tmp/allhosts
sp2n01
sp2n05
sp2n06
   .
   .
   .
sp5en0
sp5n01
sp5n05
^C
```

Next we set the WCOLL variable to this file and export it:

```
# export WCOLL=/tmp/allhosts
```

It is well known that the `dsh` command without any option takes the contents of the file to which the WCOLL variable points:

```
# dsh date
```

But how can you use this WCOLL variable with other parallel commands, such as `pcp`, `lppdiff`, `pdf`? The dsh command without any options is actually a short form of the command:

```
# dsh '' date
```

Let us try with two nodes in the working collective:

```
sp2en0:/# cat /tmp/nodes
sp2n01
sp2n05
sp2en0:/# export WCOLL=/tmp/nodes
sp2en0:/# dsh '' date
sp2n01: Sun Apr 11 21:12:34 EDT 1999
sp2n05: Sun Aug 11 21:12:34 EDT 1999
sp2en0:/# pcp '' /tmp/blubb
sp2en0:/# dsh '' ls -la /tmp/blubb
sp2n01: -rw-r--r--   1 root     system         6 Apr 02 21:13 /tmp/blubb
sp2n05: -rw-r--r--   1 root     system         6 Apr 02 21:13 /tmp/blubb
sp2en0:/# lppdiff '' ssp.css
-------------------------------------------------------------------------------
      Name         Path                Level        PTF     State      Type  Num
-------------------------------------------------------------------------------
LPP:  ssp.css      /etc/objrepos       3.1.0.0              COMMITTED  I       2
From: sp2n01 sp2n05
-------------------------------------------------------------------------------
LPP:  ssp.css      /usr/lib/objrepos   3.1.0.0              COMMITTED  I       2
From: sp2n01 sp2n05
sp2en0:/# pdf '' /var
Filesystem         Size-KB    Used-KB    Free-KB %Free    iUsed    iFree %iFree
-----------------  -------    -------    ------- -----    -----    ----- ------
HOST: sp2n01
------------
/var                 32768       9148      23620   73%      368     7824    96%

HOST: sp2n05
------------
/var                  4096       4068         28    1%      338      686    67%
```

This is how all the other mentioned commands can work with the WCOLL variable.

Chapter 9. Integration of External Workstations

With PSSP 3.1, a new strategy has arisen. The High Availability Infrastructure that was restricted to the SP Environment before now becomes a Cluster Technology, spread over a cluster consisting of both SP systems and external machines. Further, HACMP Enhanced Scalability Version 4.3 moves beyond partition boundaries and can be used over several partitions and different SP systems, as well as external workstations.

Because of these major changes, it makes sense to integrate external workstations with some services offered by the CWS. This chapter covers the integration of external workstations into the SP Kerberos realm, as well as the definition of external workstations to the NIM master configuration on the CWS.

9.1 Integration of External Workstations to the SP Kerberos Realm

For security reasons it may be required to integrate RS/6000 systems into a SP Kerberos realm. In this section we show how to integrate an external workstation into a existing Kerberos realm.

Kerberos never sends a password unencrypted over the network. That is fine, but as an SP administrator, you usually do not work directly in the lab but instead telnet from another machine to your CWS by typing the root password. If you authenticate to the realm, you will be asked for the root.admin password. During both authentication procedures, the password between your workstation and the CWS is transferred across the network unencrypted. So, to maintain this security function, it makes sense to integrate at least your workstation to your SP Kerberos realm.

The procedure to do this is nearly the same as merging two SPs into one Kerberos realm. In the following section we describe the necessary steps required for realm integration. For background information concerning the Kerberos-related files, the rcmd principal, and the srvtab file on the nodes, refer to Chapter 8, “Taming Kerberos” on page 155.

In our example we will show the integration of the machine named risc77 that is connected to the CWS over Token Ring. Figure 41 on page 206 gives an overview of our scenario.

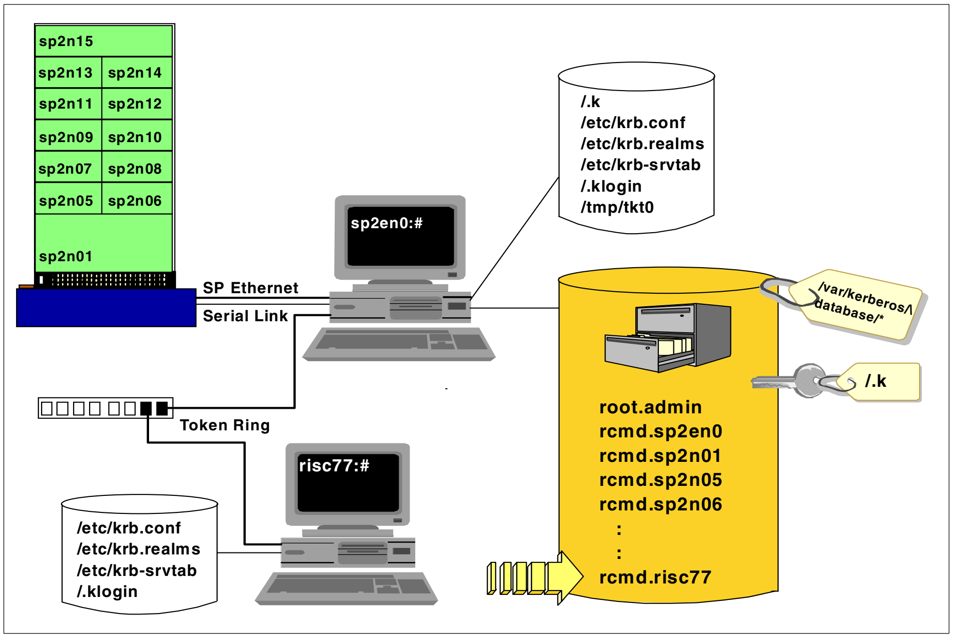


Figure 41. External Workstation as Part of the SP Kerberos Realm

9.1.1 Time Synchronization, Name Resolution and Kerberos Code

Kerberos packets have a maximum lifetime of five minutes. If the time gap between two machines exceeds this limit, Kerberos authentication and communication will not work. The SP uses the Network Time Protocol to achieve time synchronization between all participants. For your external machine, you have to insure that there is no time difference between the SP system and your external workstation. This can be done by using a time server that serves both the CWS and your external workstations.

Make sure that the name resolution for your external workstation works properly. We recommend that you include the workstation in the /etc/hosts file on your CWS and also include the CWS and the node in the /etc/hosts file on your workstation.

To obtain the necessary Kerberos environment and commands on your external workstation, you have to install the Kerberos client code, meaning the ssp.clients fileset of the PSSP software. You can mount the spdata/sys1/install/pssplpp directory from the CWS and install the ssp.fileset from the appropriate PSSP-<code.version> subdirectory. After the installation, the `lslpp` command should return the following:

```
risc77:/# lslpp -L | grep ssp
ssp.clients               3.1.0.0    C    (ECIP) SP Authenticated Client
```

9.1.2 Edit /etc/krb.realms on the Control Workstation

The /etc/krb.realms file indicates which host, or better, which adapter name, belongs to the realm. Therefore, you have to add the adapter name of your external workstation to the /etc/krb.realms file on the CWS. In our example we add the risc77 adapter name to that file.

```
sp2en0:/# tail /etc/krb.realms
sp2css11 SP2EN0
sp2n12 SP2EN0
sp2css12 SP2EN0
sp2n13 SP2EN0
sp2css13 SP2EN0
sp2n14 SP2EN0
sp2css14 SP2EN0
sp2n15 SP2EN0
sp2css15 SP2EN0
risc77 SP2EN0
```

9.1.3 Extend the /.klogin File on the Control Workstation

When a Kerberos principal tries to set up a remote shell command on a remote host, the last step of authentication is the check of the /.klogin file on the remote host. If the principal that asks for the service is not found in the /.klogin file, the service is refused. Currently the root.admin principal is included to the /.klogin file. Later this file will be distributed to the external workstation. You can include the rcmd principal of the external workstation in the /.klogin file, but it is not mandatory. What would be the difference?

As long as you are authenticated as the Kerberos principal root.admin either on host sp2en0 or on risc77, the /.klogin check will return okay. But as soon as you use the `rcmdtgt` command on risc77, which gives you a ticket for the rcmd.risc77 principal, you will not pass the /.klogin check because the rcmd.risc77 principal is not part of the /.klogin.

In any case the integration of the rcmd.principal avoids authentication errors. We will include the rcmd principal for risc77 in the /.klogin file on the CWS, as follows:

```
# cat /.klogin | pg
root.admin@SP2EN0
rcmd.sp2en0@SP2EN0
rcmd.risc77@SP2EN0
rcmd.sp2n01@SP2EN0
rcmd.sp2n05@SP2EN0
rcmd.sp2n06@SP2EN0
rcmd.sp2n07@SP2EN0
rcmd.sp2n08@SP2EN0
rcmd.sp2n09@SP2EN0
^C
```

9.1.4 Add rcmd Principals to the Kerberos Database

As described in 8.11.5, “Adding Principals for Remote Nodes” on page 193, you have two possibilities for adding a principal to the Kerberos database. You can use the `kadmin` command or you can use the `add_principal` command. The `add_principal` command needs a file as input in which the principal and a password is provided. For just one principal, it is not worth creating a file, so we will define the rcmd principal by using `kadmin` command. If you are interested in the other method, see 8.11.5, “Adding Principals for Remote Nodes” on page 193.

```
p2en0:/# kadmin
Welcome to the Kerberos Administration Program, version 2
Type "help" if you need it.
admin:  ank rcmd.risc77
Admin password:
Password for rcmd.risc77:
Verifying, please re-enter Password for rcmd.risc77:
rcmd.risc77 added to database.
admin:  quit
Cleaning up and exiting.
```

Repeat this action for each adapter configured in the external workstation. Just add a principal rcmd.<adapter name> to the database. First you are prompted for the admin password that is the root.admin password, then you are prompted for the rcmd.principal’s password. Provide a password. Choose an arbitrary one and bear in mind that you will never need it again. Why? Because you get authenticated as an rcmd principal by issuing `rcmdtgt` and this command does not ask you for any password. For details refer to 8.7.6, “Never-Expiring Ticket” on page 176.

9.1.5 Create a krb-srvtab File for the External Workstation

As soon as the rcmd principal is defined, a srvtab file can be extracted from the Kerberos database containing the password for this principal. Later this file will be stored under /etc/krb-srvtab on the external workstation. Since we just include one adapter name of the external workstation, there is only one srvtab to extract. If you extract several srvtabs due to having several rcmd principals, these srvtab files have to be concatenated to one file (refer to 8.10, “Recreating the krb-srvtab File for a Node” on page 187).

The syntax of the command looks like the following:

```
# ext_srvtab [-n] [-r realm] instance [instance...]
```

```
sp2en0:/tftpboot# ext_srvtab -n risc77
Generating 'risc77-new-srvtab'....
sp2en0:/tftpboot# ls -la | grep risc77
-rw-------   1 root     system       28 Aug  5 13:49 risc77-new-srvtab
```

9.1.6 Transfer the Required File to the External Workstation

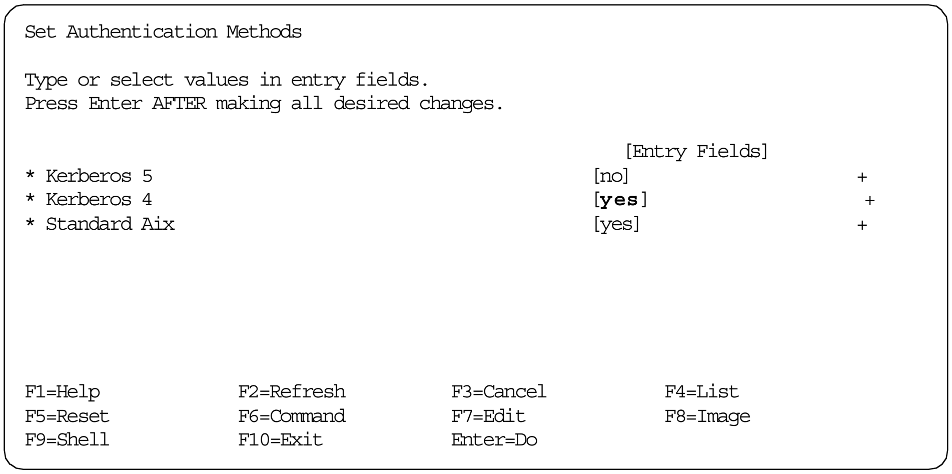
Next, the required Kerberos files have to be transferred to the external workstation; these are /etc/krb.conf, /etc/krb.realms, and /.klogin. And last but not least, the just-created srvtab has to be copied to the workstation as a /etc/krb-srvtab file. We use `ftp` and just show the commands (the messages have been truncated in this example).

```
sp2en0:/tftpboot# ftp risc77
ftp> bin
ftp> put /etc/krb.conf
ftp> put /etc/krb.realms
ftp> put /.klogin
ftp> put /tftpboot/risc77-new-srvtab /etc/krb-srvtab
ftp> quit
```

9.1.7 Set the Authentication Method on the External Workstation

With AIX 4.3.1, an authentication option is provided. There is a choice as to which kind of authentication will be used. We use Kerberos version 4 on our external workstation. This has to be defined; otherwise, the authentication services will not work even if all files are provided and correct. Set the authentication method by using SMIT.

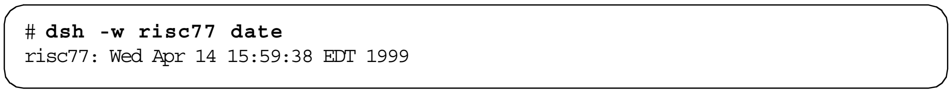
1. Type `smitty auth_set`.
2. Choose Kerberos 4.



```
Set Authentication Methods

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Kerberos 5                                         [no]                    +
* Kerberos 4                                         [yes]                   +
* Standard Aix                                       [yes]                   +







F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

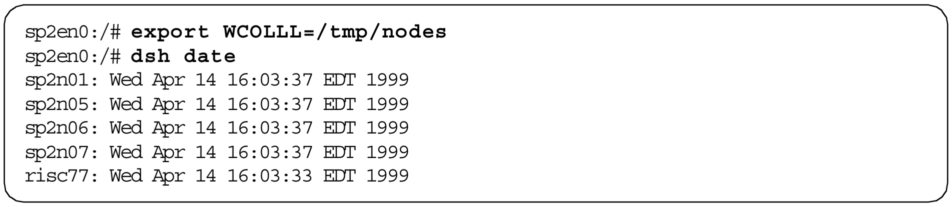
If you set Kerberos 4 to yes but set Standard AIX and Kerberos 5 to no, standard remote login (telnet, tfp) is not longer provided. Kerberos 5 requires DCE version 2.2 or higher.

After enabling Kerberos 4 as an authentication method, the kerberized remote shell commands issued from the CWS to our external workstation should work.



```
# dsh -w risc77 date
risc77: Wed Apr 14 15:59:38 EDT 1999
```

Including this external workstation to the file that is referenced by the WCOLL variable (in our example, /tmp/nodes) makes this workstation accessible in combination with your SP nodes.



```
sp2en0:/# export WCOLLL=/tmp/nodes
sp2en0:/# dsh date
sp2n01: Wed Apr 14 16:03:37 EDT 1999
sp2n05: Wed Apr 14 16:03:37 EDT 1999
sp2n06: Wed Apr 14 16:03:37 EDT 1999
sp2n07: Wed Apr 14 16:03:37 EDT 1999
risc77: Wed Apr 14 16:03:33 EDT 1999
```

9.2 Using the CWS NIM Master Setup to Install External Workstations

Since you have a NIM Master Setup on your CWS anyway, why not use this NIM configuration for external workstations that are not part of the SP environment? Almost all needed resources are already created, such as lppsource, spot, boot and so on. Only the definition of the external NIM clients and the network over which they will get installed are missing. In this section, we will show what steps remain in order to include external workstations to your NIM configuration on the CWS. For detailed information about NIM and NIM on the SP, refer to 2.1, “NIM Overview” on page 9.

9.2.1 Definition of a Second Network Served by the NIM Master

A NIM configuration always consists of a NIM master in combination with at least one network over which the installations take place. Moreover, there are NIM client definitions that also include information on which client uses which network for installation. Finally, there are the definitions of resources that are necessary during installation, migration and maintenance. All the previously mentioned components are already available on your CWS, with the internal SP Ethernet as the defined network for NIM procedures.

Since you cannot use the internal SP Ethernet for the installation of external clients, you have to define a second network served by the NIM master.

In regard to NIM network definitions on a CWS, although it is physically possible to connect external workstations to the internal SP Ethernet, it is not supported to extend the SP Ethernet and connect other clients. For non-SP clients, you have to define a second network served by the master. However, one problem still remains: as soon as `setup_server` runs the next time, the definitions of non-SP clients will be deleted. The `setup_server` command maintains the whole NIM environment and uses the SDR as input. Consequently, it detects that this client is “no longer” part of the SDR, so the definition will be removed.

Until you migrate to this level, however, there is a workaround. Since only the client definition is deleted (no network or resource is deleted), it is very easy to write a script that redefines the non-SP clients after `setup_server` runs. Figure 42 on page 212 shows an example configuration and we provide you with the workaround script.

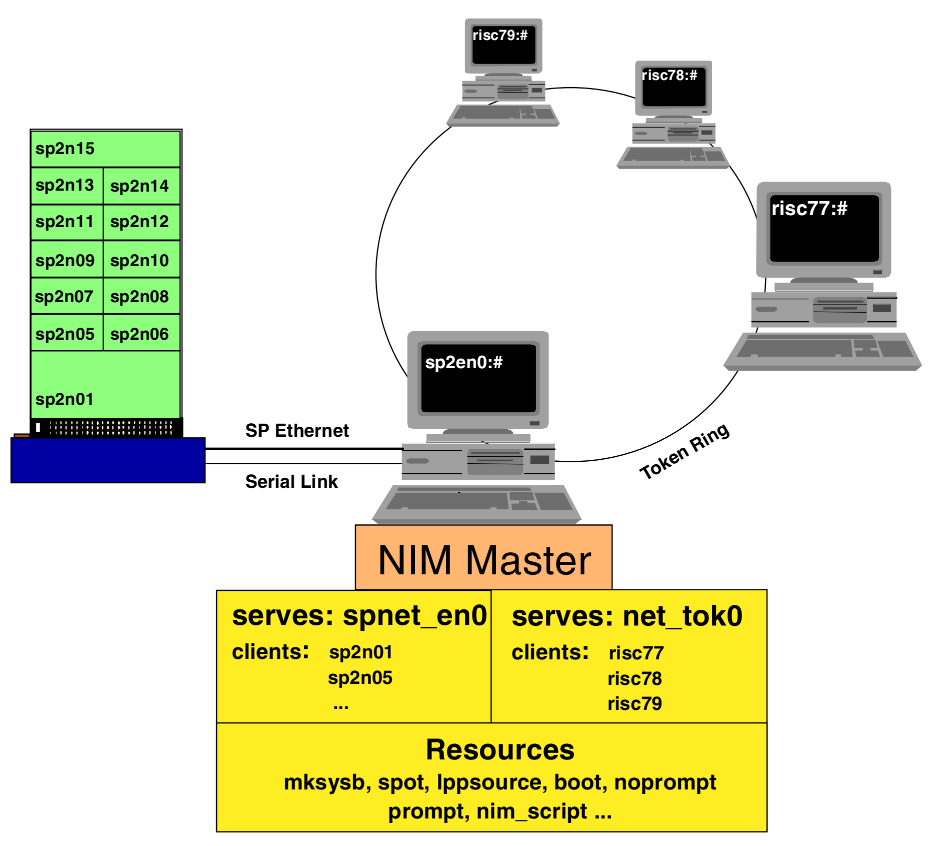
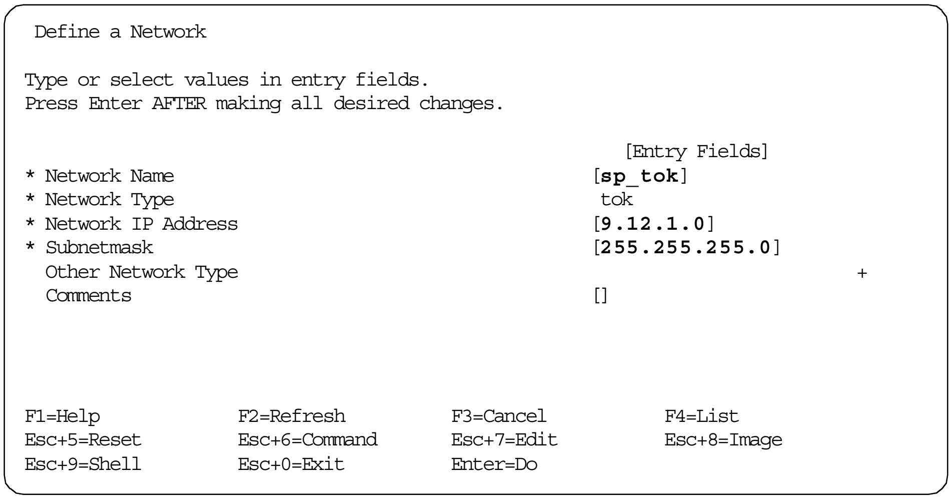


Figure 42. The CWS as NIM Master for a Second Network

Since the external hosts are not part of the SP, we cannot use the helpful wrappers of `setup_server` . Instead, we are forced to use the standard NIM commands. In our example we are going to install the risc77 from the NIM master sp2en0 over the Token Ring. First we define the second network. The SMIT fastpath for NIM administrative tasks is nim_mknet.

1. Type `smitty nim_mknet`.
2. Select the appropriate network interface.

You will see the following menu:



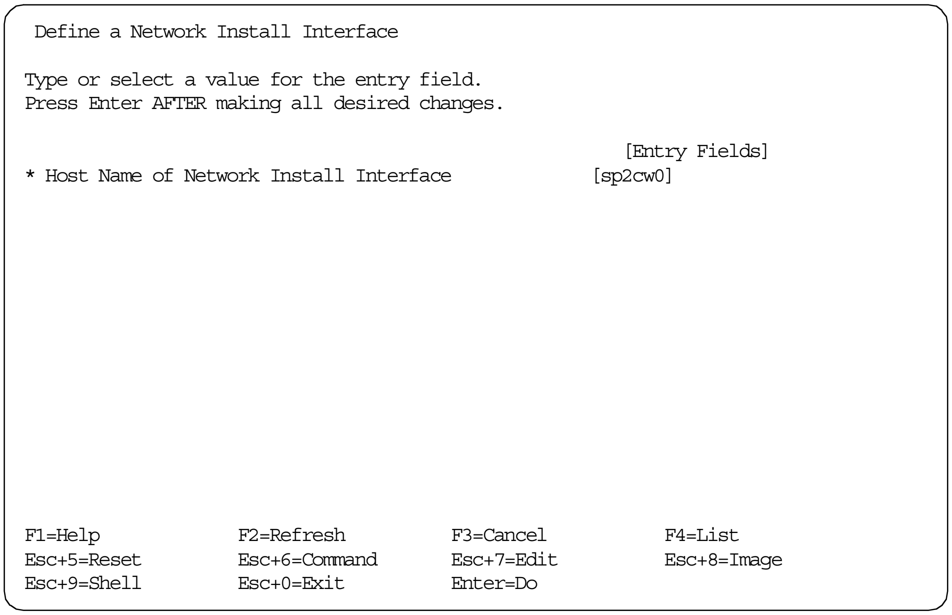
```
 Define a Network

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* Network Name                                      [sp_tok]
* Network Type                                       tok
* Network IP Address                                [9.12.1.0]
* Subnetmask                                        [255.255.255.0]
  Other Network Type                                                        +
  Comments                                          []




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

The fields that are required are shown in bold characters. The Network Name is free-choosable, and it is the equivalent value to spnet_en0.

Up to now only the network itself is defined, so this new network now has to be defined as an interface served by the NIM master, as follows:

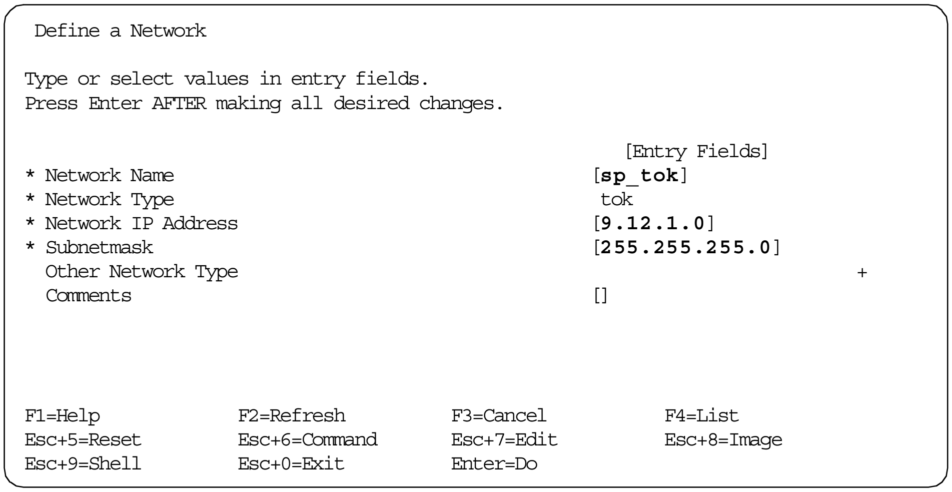
1. Type `smitty nim_mkmac_if`.
2. Select **master.**
3. Enter the appropriate interface.

You will see the following menu:



```
 Define a Network Install Interface

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Host Name of Network Install Interface             [sp2cw0]

























F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Select the appropriate network interface (Token Ring, in our example). Then you will see the following screen.



```
 Define a Network

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Network Name                                       [sp_tok]
* Network Type                                        tok
* Network IP Address                                 [9.12.1.0]
* Subnetmask                                         [255.255.255.0]
  Other Network Type                                                         +
  Comments                                           []




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

The Hardware Address of the interface is needed, and you can get it with the command:

```
# lscfg -v -l tok0
```

Now we check to see if the master serves both interfaces:

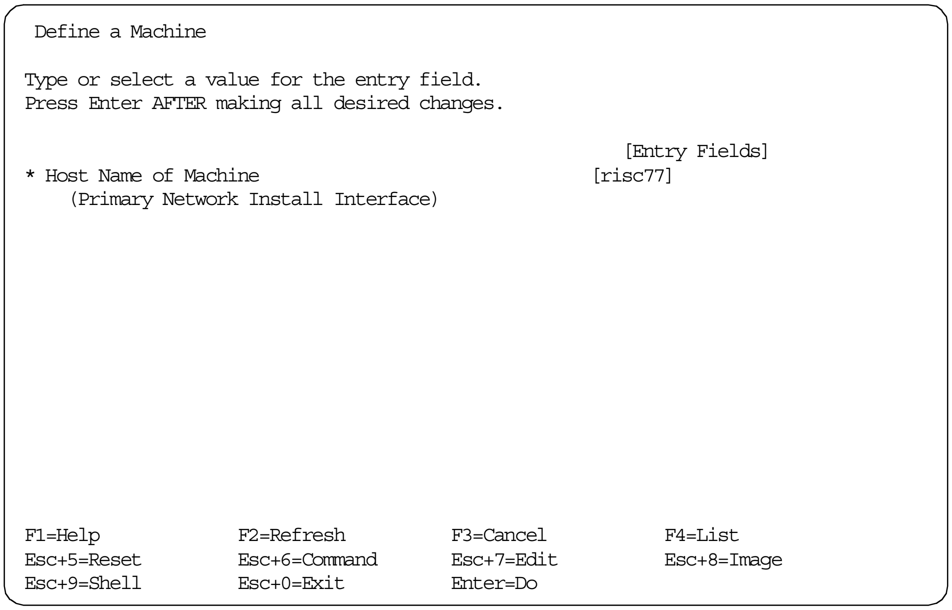
```
# lsnim -l master| grep if[0-9]
if1             = spnet_en0 sp2en0 02608C2D4A7F
if2             = sp_tok sp2cw0 10005AB1919C tok
```

9.2.2 External NIM Client Definition

Which kind of client will the external workstation be? Since we are working with the standard NIM functions independent of `setup_server` and the SDR, we have the entire NIM functionality. Nevertheless, we will define the external workstation as a standalone client like the SP nodes.

Type `smitty nim_mkmac`.

You will see the following menu:



```
 Define a Machine

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Host Name of Machine                               [risc77]
    (Primary Network Install Interface)















F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

After entering the hostname of the external NIM client, you will see the following screen.

```
 Define a Machine

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* NIM Machine Name                                   [risc77]
* Machine Type                                       [standalone]            +
* Hardware Platform Type                             [rs6k]                  +
  Kernel to use for Network Boot                     [up]                    +
  Primary Network Install Interface
*   Ring Speed                                       [16]                     +
*   NIM Network                                       sp_tok
*   Host Name                                         risc77
    Network Adapter Hardware Address                 [10005AA8E7A3]
    Network Adapter Logical Device Name              [tok]
  IPL ROM Emulation Device                           []                      +/
  CPU Id                                             []
  Machine Group                                      []                      +
  Comments                                           []


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

In this menu you are not required to fill in the CPU ID of the client. After the first installation, the NIM master fetches the CPU ID of the client itself. The following screen shows the client definition.

```
sp2en0:/# lsnim -l risc77
risc77:
   id             = 902176487
   class          = machines
   type           = standalone
   platform       = rs6k
   netboot_kernel = up
   if1            = sp_tok risc77 10005AA8E7A3 tok
   ring_speed1    = 16
   Cstate         = ready for a NIM operation
   prev_state     = ready for a NIM operation
   Mstate         = currently running
```

As already mentioned, this external client definition will be removed during the next `setup_server` run. As a workaround until this is fixed, you can write a shell script that defines the client. The manual definition of our NIM client risc77 would look like this:

```
# nim -o define -t standalone -a if2="sp_tok risc77 \
10005AA8E7A3 tok" -a ring_speed=16 -a platform=rs6k \
-a netboot_kernel=up risc77
```

9.2.3 Creating the Resources for External NIM Clients

All NIM resources available on a CWS can be used for external clients except for the script resource that is used for SP nodes, because neither an ssp fileset installation nor any SP-related customization has to run at the end of an external client installation. Nevertheless, it is possible to write another customization script for other clients and define it as a NIM resource. Once the client’s installation has started and there is a script resource allocated for it, this script runs at the end of the installation on the client. For details on the script resource, see 2.1.5, “NIM Resources” on page 17. In our example, we do not use a script resource.

Finally, an mksyb resource has to be defined pointing to the mksysb of our workstation risc77. We placed one under /spdata/sys1/install/images. However, you can put it wherever you want, but you should insure that this file can be exported. Once the mksysb file is placed somewhere, you can create the mksysb resource by using SMIT:

1. Type `smitty nim_mkres`.
2. Choose **mksysb.**

You should see the following menu:

```
 Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
* Resource Name                                     [mksysb_risc77]
* Resource Type                                      mksysb
* Server of Resource                                [master]           +
* Location of Resource                             <images/risc77.img]  /
  Comments                                          []

  System Backup Image Creation Options:
    CREATE system backup image?                      no                +
    NIM CLIENT to backup                            []                 +
    PREVIEW only?                                    no                +
    IGNORE space requirements?                       no                +
    EXPAND /tmp if needed?                           no                +
    Create MAP files?                                no                +
[MORE...7]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

After that definition, `lsnim` returns the following screen.

```
sp2en0:/# lsnim -l mksysb_risc77
mksysb_risc77:
   class       = resources
   type        = mksysb
   Rstate      = ready for use
   prev_state  = unavailable for use
   location    = /spdata/sys1/install/images/risc77.img
   version     = 4
   release     = 3
   mod         = 1
   alloc_count = 0
   server      = master
```

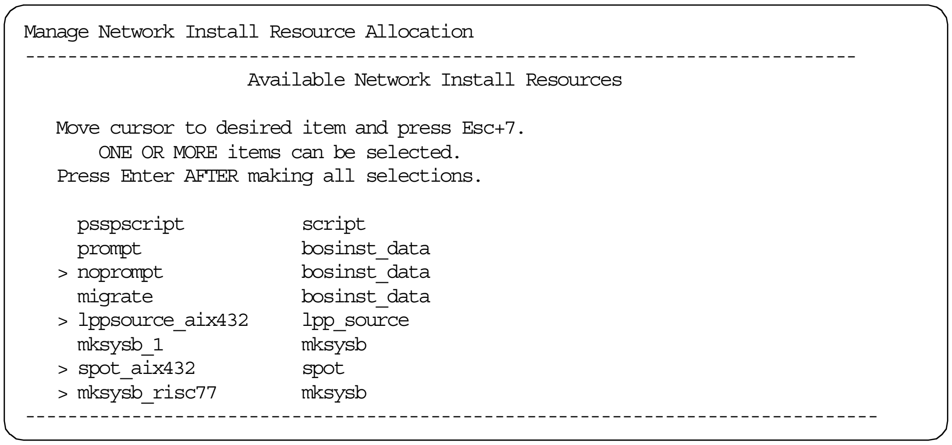
Now the additional definitions for our external workstations on the NIM master are finished completed.

9.2.4 Setting Up the Installation of an External Workstation

Unfortunately `setup_server` cannot be used for the allocation of the necessary resources for an external client installation. We have to set it up manually. However, it is helpful to use the `allnimres` script (`setup_server` wrapper) to look up the appropriate commands for the resource allocation and preparation and write your own setup script for the external clients. In our example we use SMIT.

1. Type `smitty nim_mac_res`.
2. Select **Allocate Network Resources.**
3. Choose the external NIM client.

You should see the following menu:



```
Manage Network Install Resource Allocation
---------------------------------------------------------------------------
                    Available Network Install Resources

   Move cursor to desired item and press Esc+7.
       ONE OR MORE items can be selected.
   Press Enter AFTER making all selections.

     psspscript            script
     prompt                bosinst_data
   > noprompt              bosinst_data
     migrate               bosinst_data
   > lppsource_aix432      lpp_source
     mksysb_1              mksysb
   > spot_aix432           spot
   > mksysb_risc77         mksysb
-----------------------------------------------------------------------------
```

Select the needed resources, depending upon to your setup. In our example they are noprompt, lppsource_aix432, spot_aix432 and mksysb_risc77.

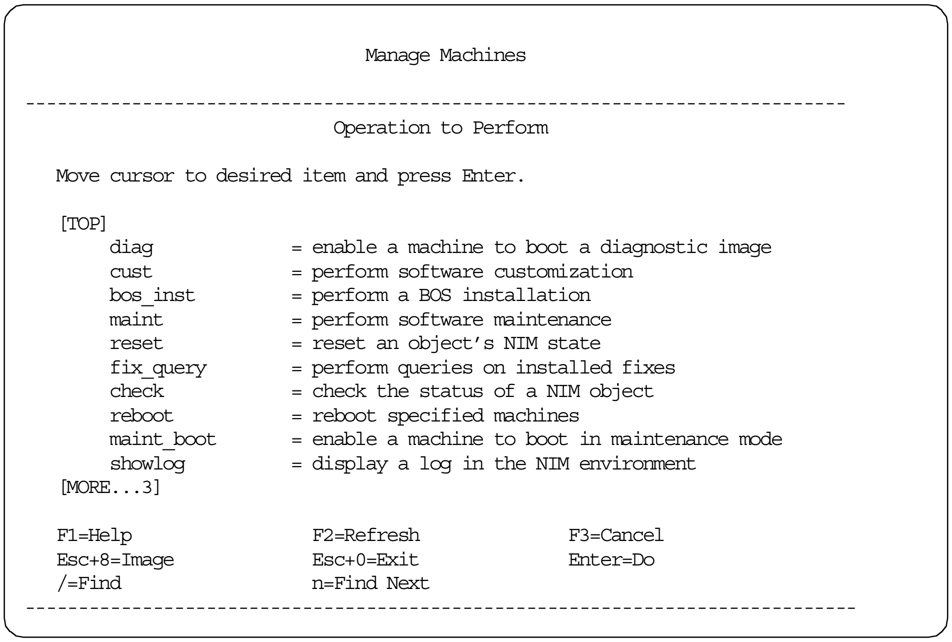
Finally, the boot resource has to be allocated, the required directories have to be exported, and an entry to the /etc/bootptab has to be made for our external NIM client. This is done by the command:

```
# nim -o bosinst <nim_client>
```

If you go through the SMIT panels, this is done by the command:

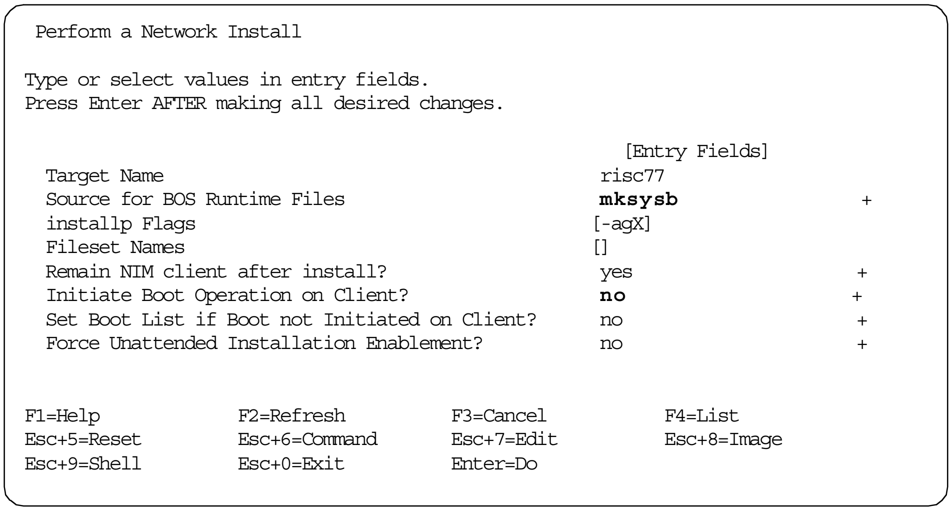
1. Type `smitty nim_mac`.
2. Select **Perform Operations on Machines.**
3. Choose the extern NIM client.

You should see the following menu:



```
                              Manage Machines

-----------------------------------------------------------------------------
                             Operation to Perform

   Move cursor to desired item and press Enter.

   [TOP]
        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
        fix_query       = perform queries on installed fixes
        check           = check the status of a NIM object
        reboot          = reboot specified machines
        maint_boot      = enable a machine to boot in maintenance mode
        showlog         = display a log in the NIM environment
   [MORE...3]

   F1=Help                 F2=Refresh              F3=Cancel
   Esc+8=Image             Esc+0=Exit              Enter=Do
   /=Find                  n=Find Next
------------------------------------------------------------------------------
```

Choose **bos_inst** and fill out the next screen:



```
 Perform a Network Install

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
  Target Name                                           risc77
  Source for BOS Runtime Files                          mksysb              +
  installp Flags                                       [-agX]
  Fileset Names                                        []
  Remain NIM client after install?                      yes                 +
  Initiate Boot Operation on Client?                    no                  +
  Set Boot List if Boot not Initiated on Client?        no                  +
  Force Unattended Installation Enablement?             no                  +


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

With these choices, you have to start the client installation manually and the mksyb image that is allocated will be used. Before starting the installation, the allocated NIM resources, the /tftpboot/<client_name> file and the

/etc/bootptab file can be verified. We will just have a look into these files, as shown by the following screen, but for detailed information, see 2.1.2, “NIM Master” on page 14.

```
sp2en0:/# lsnim -c resources risc77
noprompt            bosinst_data
lppsource_aix431    lpp_source
risc77_mksysb       mksysb
spot_aix431         spot
boot                boot              represents the network boot resource
nim_script          nim_script        directory containing customization scrip
sp2en0:/# tail /etc/bootptab
#      T175 -- (xstation only) -- primary / secondary boot host indicator
#      T176 -- (xstation only) -- enable tablet
#      T177 -- (xstation only) -- xstation 130 hard file usage
#      T178 -- (xstation only) -- enable XDMCP
#      T179 -- (xstation only) -- XDMCP host
#      T180 -- (xstation only) -- enable virtual screen
risc77:bf=/tftpboot/risc77:ip=9.12.1.77:ht=token-ring:ha=10005AA8E7A3:
sa=9.12.1.37:sm=255.255.255.0:
sp2en0:/# ls -la /tftpboot/risc77
lrwxrwxrwx   1 root     system       33 Aug  5 17:14 /tftpboot/risc77 -> /tftpb
oot/spot_aix431.rs6k.up.tok
```

All entries are correct. The necessary resources are allocated: there is one entry in the /etc/bootptab file for the client risc77 and the /tftpboot/risc77 file points to the appropriate uniprocessor boot kernel, indicated as bootfile (bf) in the bootptab entry.

9.2.5 Initializing the Client Installation

When the client resides in the same network as the NIM server, you can change the bootlist of the client to the appropriate interface. As soon as you reboot the client machine, it will try to boot over the network by sending out bootp requests. The NIM master will react with a bootp response because there is an entry in the /etc/bootptab file for this client. The command for our external NIM client is:

```
# bootlist -m normal tok0
```

A second way to initialize the client installation is to use the IPL ROM of the client machine. It will offer you, in a menu, the interfaces that are available and you can choose over which interface it should boot. If you have already done the manual node conditioning on an SP node, you will be familiar with this procedure:

1. Set key to secure.
2. Power on and wait for LED 200.

3. Set key to service and reset.

At this point, the IPL ROM menu will be offered. You can then make the appropriate choices and start the system boot over the network.

Chapter 10. The Switch

After the installation of the nodes, you might start the Switch interface. Usually this is done from the CWS with the `Estart` command, and you hope that every node will join the switch. However, several things can occur between issuing `Estart` and getting the green switch responds: It can fail at once; or only some of the nodes join the switch; or the primary node is fenced and when you try to unfence it, you get a message that it cannot be unfenced because it is the primary node, and so on.

To avoid these situations, there are several verification steps that you can do before using `Estart`. This chapter describes these verification steps.

10.1 Definition of the Switch

Between running `setup_server` and starting the installation of the nodes over the net, you define the Switch by annotating a topology file, defining the primary node and primary backup node, and setting the switch clock source, which means you simply add definitions to the System Data Repository (SDR). Verification of these definitions can be done by several commands.

10.1.1 Checking the Primary and Primary Backup Node

There are three types of nodes in a Switch environment:

1. Primary node

   This node initializes the switch, distributes topology, fences and unfences nodes and updates the SDR.

2. Primary backup node

   This node monitors the primary nodes, and if the primary fails, it becomes the primary node and defines the next primary backup.

3. Secondary node

   This refers to every node that is neither the primary nor primary backup node.

An SP Switch Board contains eight SP Switch chips that provide connections for each of the nodes to the SP Switch network. Each SP Switch chip has 8 Switch ports which are used for data transmission. Figure 43 on page 224 gives an overview of the connections within an SP Switch board.

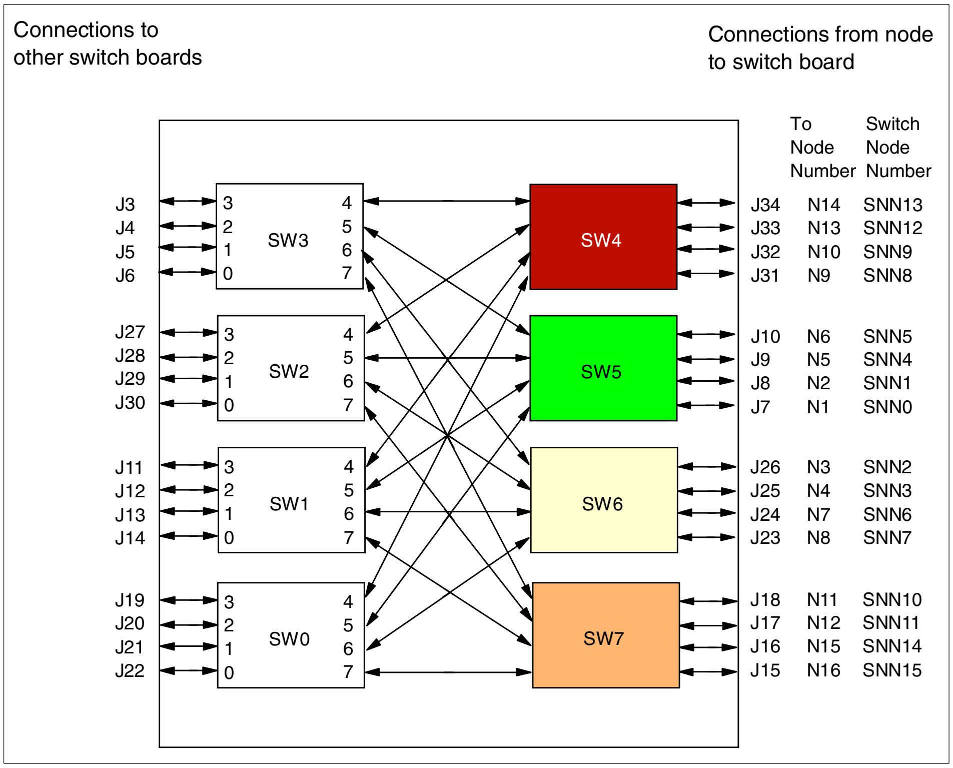


Figure 43. SP Node Switch Board

The rule for setting primary and primary backup node is as follows: if you have more than one switch board, spread them over different switch boards. But in any case, do not define the primary and primary backup on nodes that share the same switch chip because if a switch chip fails, the primary backup node will not be able to become the primary node (see 10.2.2, “The Worm Daemon” on page 230).

Figure 44 on page 225 shows the same topic, but focuses on the nodes in the frame.

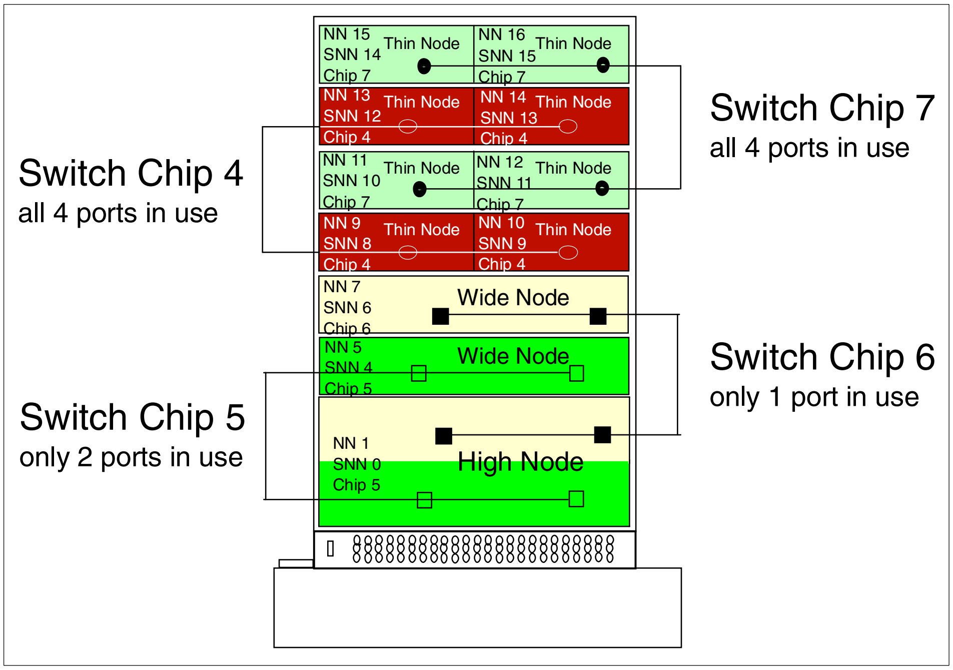


Figure 44. SP Node - Switch Chip Connections

In our configuration, five of the 16 available switch ports are left because only nine nodes are connected to the switch board. They could be used, for example, to connect the SP Switch Router or S70 nodes.

The `Eprimary` command without options returns the current settings for the primary and primary backup node. With the old High Performance Switch (HiPS), the output of the `Eprimary` command is only one line telling you which node is currently the primary node. For more information about the differences between HiPS and SPS, refer to *RS/6000 SP: Problem Determination Guide,* SG24-4778, Chapter 4.

The oncoming primary and oncoming primary backup node will become the primary and primary backup node, respectively, after the next `Estart`. In our example, the Switch is currently not up, therefore primary and primary backup are marked as *none*.

```
 Eprimary
none    - primary
1       - oncoming primary
none    - primary backup
7       - oncoming primary backup
```

If the Switch were up and running, the output would be the following:

```
 Estart
1       - primary
1       - oncoming primary
7       - primary backup
8       - oncoming primary backup
```

When you set the primary or the primary backup node (or both) to another node, this only changes the oncoming values stored in the SDR. These values will be activated during the next `Estart`.

10.1.2 Checking the Switch Clock Source Settings

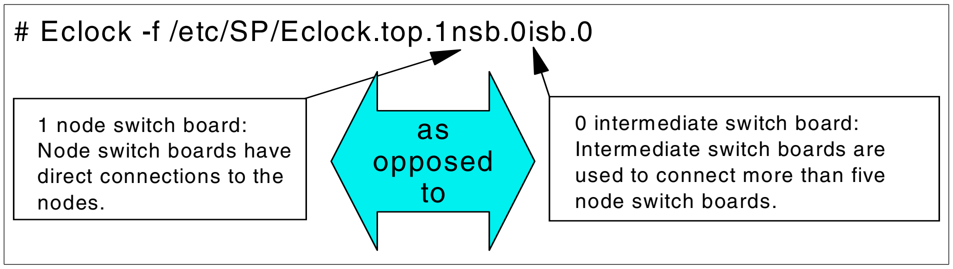
With the `splstdata -s` command, you get information about switch node numbers, which node is connected to which switch chip, over which port it communicates and so on. In the second part of the output, the definition of the switch clock source setting in the SDR is returned.

In a one-frame configuration, there is only one possibility for the switch clock source setting. The clock input has to be 0, meaning the internal clock of this switch board is used.

In a multiframe configuration, there has to be one dedicated board that serves the clock signal to the others. In our example we have a two-frame SP and you can see that the clock input for switch board 1 is 0 (internal clock), but for switch board 2 the clock input is 1. This board is getting the clock signal from switch board 1.

```
# splstdata -s
 List Node Switch Information

                        switch  switch  switch switch    switch
node# initial_hostname   node# protocol number   chip chip_port
----------------------------------------------------------------
    1 sp4n01                 0       IP      1      5         3
    5 sp4n05                 4       IP      1      5         1
    6 sp4n06                 5       IP      1      5         0
    7 sp4n07                 6       IP      1      6         2
    8 sp4n08                 7       IP      1      6         3
    9 sp4n09                 8       IP      1      4         3
   13 sp4n13                12       IP      1      4         1
   14 sp4n14                13       IP      1      4         0
   15 sp4n15                14       IP      1      7         2
   17 sp4n17                 0       IP      2      5         3
   19 sp4n19                 2       IP      2      6         0

switch   frame    slot  switch_partition  switch  clock switch
number  number  number            number    type  input  level
----------------------------------------------------------------
     1       1      17                 1     129      0
     2       2      17                 1     129      1

switch_part            topology          primary        arp  switch_node
     number            filename             name    enabled    nos._used
-------------------------------------------------------------------------
          1   expected.top.an   sp4n01                  yes           no
```

Since this is just the definition of the clock source setting in the SDR, it may happen that the switch clock signal is missing on the board or on a switch adapter. In this case, you are not able to start the Switch and you will find an entry in the error report. To set the clock source again, look for the appropriate Eclock file fitting to your configuration and issue the command shown in Figure 45:



```
# Eclock -f /etc/SP/Eclock.top.1nsb.0isb.0
```

1 node switch board: Node switch boards have direct connections to the nodes.
as opposed to
0 intermediate switch board: Intermediate switch boards are used to connect more than five node switch boards.

Figure 45. Eclock

This command should only be used when necessary, because it stops the entire Switch.

10.2 Initialize the Switch

The start of the switch interface can be divided into three parts:

1. The `Estart` command on the CWS
2. The `Estart_sw` script running on the oncoming primary node
3. Computing of routes and downloading to the switch adapter on each node

In Figure 46 we can see what is going on during switch initialization.

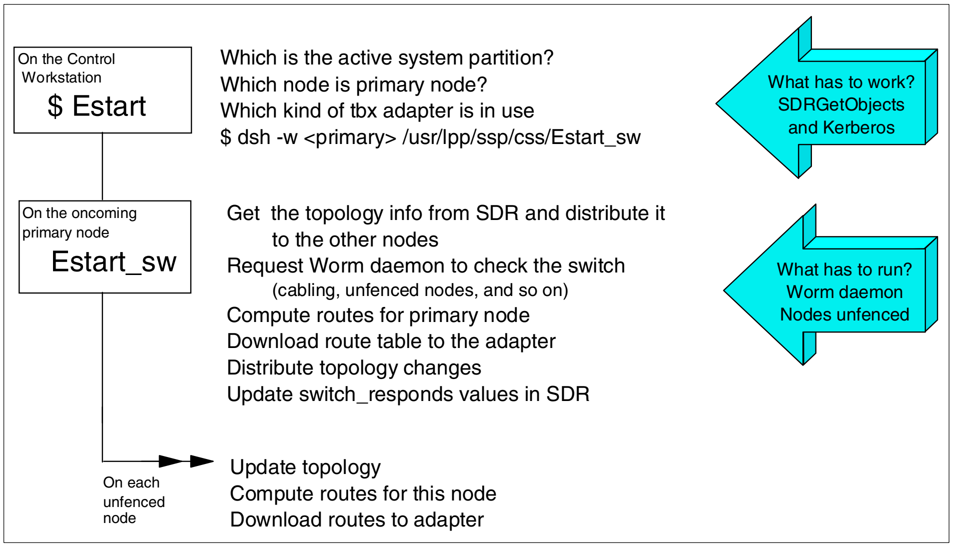

On the Control Workstation
$ Estart
Which is the active system partition?
Which node is primary node?
Which kind of tbx adapter is in use
$ dsh -w <primary> /usr/lpp/ssp/css/Estart_sw
What has to work?
SDRGetObjects
and Kerberos
On the oncoming primary node
Estart_sw
Get the topology info from SDR and distribute it
to the other nodes
Request Worm daemon to check the switch
(cabling, unfenced nodes, and so on)
Compute routes for primary node
Download route table to the adapter
Distribute topology changes
Update switch_responds values in SDR
What has to run?
Worm daemon
Nodes unfenced
On each unfenced node
Update topology
Compute routes for this node
Download routes to adapter

Figure 46. Switch Initialization

These three parts have different prerequisites and they also run on different nodes. Verifying that the `Estart` command will bring the Switch up means knowing which script is running where and what is needed for successful execution. In the following section we discuss this in detail.

10.2.1 Estart on the Control Workstation

Usually you start the switch with the `Estart` command from the CWS. It asks the SDR what the active system partition is, which nodes are oncoming primary and oncoming primary backup, and then redirects the `Estart` command to the oncoming primary node. This is done with the distributed shell command:

```
# dsh -w <oncoming_primary_name> /usr/lpp/ssp/css/Estart_sw
```

If your Kerberos setup is not correct, `Estart` fails. You will get a command response like the one in the following screen:

```
# Estart
 Estart: Oncoming primary != primary, Estart directed to oncoming primary
rshd: Kerberos Authentication Failed: Access denied because of improper credenti
als.
/usr/lpp/ssp/rcmd/bin/rsh: 0041-004 Kerberos rcmd failed: rcmd protocol failure.
trying normal rsh (/usr/bin/rsh)
rshd: 0826-813 Permission is denied.
Estart:  0028-028 Fault service worm not up on oncoming primary node, cannot Est
art : sp4n01.
```

This error message is misleading because the Worm daemon on the oncoming primary node *is* up and running. The only problem is the Kerberos authentication. The CWS was not able to start `Estart_sw` on the oncoming primary node via dsh. Estart is being issued to the primary node: sp4n01.

When a kerberized remote shell command fails, the standard `rsh` command will be tried. In other words, if you have a /.rhosts file on the oncoming primary node allowing the root user from the CWS to set up remote shell commands, `Estart` will be successful even if Kerberos does not work. Within the Kerberos error messages the `Estart` reports how many nodes have joined the Switch.

```
# Estart
 Estart: Oncoming primary != primary, Estart directed to oncoming primary
rshd: Kerberos Authentication Failed: Access denied because of improper credenti
als.
/usr/lpp/ssp/rcmd/bin/rsh: 0041-004 Kerberos rcmd failed: rcmd protocol failure.
trying normal rsh (/usr/bin/rsh)
Estart:0028-06 Estart is being issued to the primary node: sp4n01
Switch initialization started on sp4n01.
Initialized 11 node(s).
Switch initialization completed.
rshd: Kerberos Authentication Failed: Access denied because of improper credenti
als.
/usr/lpp/ssp/rcmd/bin/rsh: 0041-004 Kerberos rcmd failed: rcmd protocol failure.
trying normal rsh (/usr/bin/rsh)
```

`Estart` finds out which node is the oncoming primary node simply by asking the SDR on the CWS. Then Estart executes a `dsh` command running `Estart_sw` on the primary node. It is possible to issue `Estart` on every node that belongs to same partition as long as you have a valid Kerberos ticket for this node.

Even if you type `Estart_sw` directly on the oncoming primary node, the switch initialization will be successful, but keep in mind that in any case the SDR has to be available.

10.2.2 The Worm Daemon

The switch network function is supported by service software known as the *Worm*. This service software is coupled with the *Route Table Generator*, which examines the state of the Switch as determined by the Worm execution, and generates valid routes from every node to any other node. These two components are implemented in the fault-service daemon (fault_service_Worm_RTG_SP) which runs on every node in an SP and is central to the functioning of the switch. A diagram illustrating the relationships between the switch communication subsystem components is shown in Figure 47.

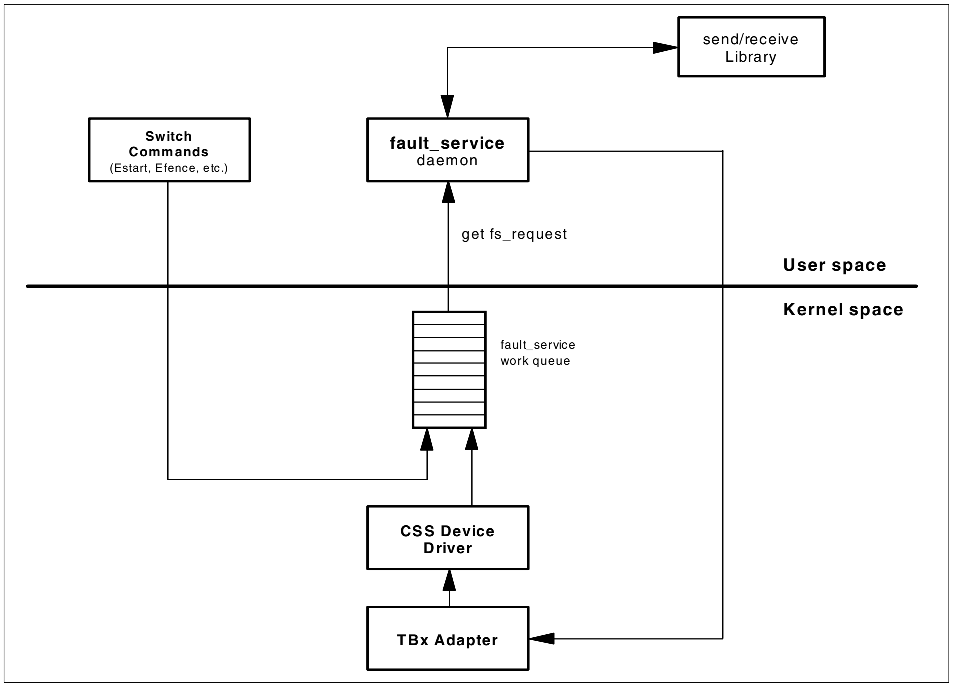


Figure 47. Switch Subsystem Component Architecture

The Worm daemon plays a key role in the coordination of the switch network. It is a non-concurrent server, and therefore can only service one switch event to completion before servicing the next. Examples of switch events (or faults) include switch initialization, Switch Chip error detection/recovery and node

(Switch Port) failure. These events are queued by an internal Worm process on the work queue. The events are serviced by the Worm when a get_fs_request system call is made to satisfy a single work queue element.

The Worm daemon is started from the /etc/inittab by the `rc.switch` script; it runs on all nodes. While the Worm daemon on every node is the same, any node is able to be or become primary or primary backup node.

Use the following command to check that Kerberos is working and the Worm daemons are running:

```
# dsh -a ps -e |grep Worm
sp4n01:  13588      -  0:00 fault_service_Worm_RTG_SP
sp4n05:  13530      -  0:00 fault_service_Worm_RTG_SP
sp4n06:  16480      -  0:00 fault_service_Worm_RTG_SP
sp4n07:  12326      -  0:00 fault_service_Worm_RTG_SP
sp4n08:  11954      -  0:00 fault_service_Worm_RTG_SP
sp4n09:  13824      -  0:00 fault_service_Worm_RTG_SP
sp4n13:  13934      -  0:00 fault_service_Worm_RTG_SP
sp4n14:  14664      -  0:00 fault_service_Worm_RTG_SP
sp4n15:   6526      -  0:05 fault_service_Worm_RTG_SP
sp4n17:  11948      -  0:00 fault_service_Worm_RTG_SP
sp4n19:  12574      -  0:00 fault_service_Worm_RTG_SP
```

If the Worm daemon is not running on a node, you can restart it by issuing the `rc.switch` script directly either on the node itself or from the CWS with a `dsh`:

```
# dsh -w <node_name> /usr/lpp/ssp/css/rc.switch
```

Sometimes it may happen that even after a restart of a node's Worm daemon, this node is not able to join the Switch, particularly if you had to set the switch clock source with the `Eclock` command. In this case you have to use a more powerful command to unload the device driver from the switch adapter, then load it again and start the Worm daemon (so `rc.switch` is not necessary), as follows:

```
$ dsh -w <node_name> /usr/lpp/ssp/css/css_restart_node
```

10.2.3 Estart_sw Running on the Oncoming Primary Node

The oncoming primary node consults the SDR to get information about the current system partition name and retrieves the appropriate topology, the oncoming primary backup node and the unfenced nodes belonging to its partition. It then distributes this topology file to these nodes.

Then the Worm is used to send packets from the primary node to each switch chip to check for cabling and to find out the routes to the primary node. Each unfenced node is sent service packets to initialize, to identify itself and to verify the switch address and readiness. The primary node updates its device database (containing the global information about the topology) in accordance with the findings and feedbacks. After the computation of the switch route table, it downloads it to the adapter and distributes the deltas of the device database to the other nodes.

10.2.4 What Happens on the Non-Primary Nodes

The unfenced non-primary nodes receive the topology file from the primary node. Then they get the deltas between the “official” SDR topology and the topology produced by the primary node (excluding, for example, the fenced nodes or the nodes that could not acknowledge to the services packet from the primary, and so on).

Based on this information, each node builds its own device database with the information on which nodes are currently ready to join the switch. Then, on each node, the Routing_Table_Generator-part (RTG) of the Worm computes the switch route table, including all the paths to the other nodes, and downloads it to the switch adapter.

Finally, the primary node updates the switch_responds class in the SDR and sets the primary and primary backup values from *none* to the appropriate nodes.

10.2.5 Fenced and Unfenced Nodes

Since Estart_sw on the oncoming primary node only cares about the unfenced nodes, fenced nodes will not be initialized during the Estart (except where the autojoin flag is set; see 10.2.6, “Fenced with and without the autojoin Flag” on page 233). The oncoming primary node retrieves the information about fenced nodes from the SDR by the following command:

```
# SDRGetObjects switch_responds
node_number  switch_responds autojoin     isolated     adapter_config_status
          1           1           0            0 css_ready
          5           1           0            0 css_ready
          6           1           0            0 css_ready
          7           0           0            1 css_ready
          8           1           0            0 css_ready
          9           0           0            1 css_ready
         13           1           0            0 css_ready
         14           1           0            0 css_ready
         15           0           0            1 css_ready
         17           1           0            0 css_ready
         19           1           0            0 css_ready
```

Currently the switch is up and running but nodes 7, 9 and 15 are isolated (perhaps they are fenced) and logically have no switch responds. For example, when you fence a node by typing `Efence 8` on the CWS, the primary node excludes this node from the switch interface, then updates its own routing table and requests all the other nodes to update their routing tables by deleting the paths to node 8. Finally, the primary node sets the isolated value in the SDR for node eight to 1. The same principle (vice versa) happens when you integrate a node by the `Eunfence` command.

Note: It is not possible to fence or unfence the primary or primary backup node.

10.2.6 Fenced with and without the autojoin Flag

Usually autojoin means the node will join the switch automatically. The autojoin value depends on the isolated value, because it is only possible to have the autojoin flag set when the node is fenced (say the isolated value equals 1.

So when will this value be evaluated? Assume the Switch is up but node 10 and 11 are already fenced, and now we want to fence two more nodes with the autojoin flag.

```
# Efence 8 9 -autojoin
```

This command isolates nodes 8 and 9 from the Switch and sets the isolated and autojoin values in the SDR for these nodes to 1. The switch responds of nodes 8 and 9 are off and the values in the SDR look like the following:

```
# SDRGetObjects switch_responds
node_number  switch_responds autojoin     isolated     adapter_config_status
          1            1            0            0 css_ready
          5            1            0            0 css_ready
          6            1            0            0 css_ready
          7            1            0            0 css_ready
          8            0            1            1 css_ready
          9            0            1            1 css_ready
         10            0            0            1 css_ready
         11            0            0            1 css_ready
         12            1            0            0 css_ready
         13            1            0            0 css_ready
         14            1            0            0 css_ready
         15            1            0            0 css_ready
```

And what is the related `spmon -d` output?

```
spmon -d
[...]
-------------------------------- Frame 1 ------------------------------------
Frame  Node    Node         Host/Switch  Key     Env   Front Panel   LCD/LED is
Slot   Number  Type  Power   Responds    Switch  Fail    LCD/LED      Flashing
-----------------------------------------------------------------------------
  1       1    high    on    yes   yes    normal   no   LCDs are blank   no
  5       5    thin    on    yes   yes    normal   no   LEDs are blank   no
  6       6    thin    on    yes   yes    normal   no   LEDs are blank   no
  7       7    thin    on    yes   yes    normal   no   LEDs are blank   no
  8       8    thin    on    yes   fence  normal   no   LEDs are blank   no
  9       9    thin    on    yes   fence  normal   no   LEDs are blank   no
 10      10    thin    on    yes   off    normal   no   LEDs are blank   no
 11      11    thin    on    yes   off    normal   no   LEDs are blank   no
 12      12    thin    on    yes   yes    normal   no   LEDs are blank   no
 13      13    thin    on    yes   yes    normal   no   LEDs are blank   no
 14      14    thin    on    yes   yes    normal   no   LEDs are blank   no
 15      15    wide    on    yes   yes    normal   no   LEDs are blank   no
```

Up to PSSP 2.4, when nodes are fenced/isolated and on autojoin, `spmon -d` returns the switch responds as *fence*.

Who cares about this value?

- `Estart`
- `rc.switch`

As described in 10.2.3, “Estart_sw Running on the Oncoming Primary Node” on page 231, `Estart` does not integrate nodes that are currently fenced. Exception: the nodes that are fenced and set to autojoin will join the Switch when `Estart` is executed.

When a node is rebooted and `rc.switch` is executed, first the Worm daemon is started and tries to talk to the Worm daemon of the primary node. If the node is only fenced, nothing will happen. However, if in addition, the autojoin value is set to 1, an `Eunfence` command for this node is executed by the primary and this will join the Switch. As soon as the switch_responds value is set to 1, the fenced and the autojoin values are set back to 0.

10.2.7 When Do You Have to Fence a Node?

While it was necessary on the old High Performance Switch to fence a node before shutting it down, on the new SP Switch this is unnecessary. The SP Switch introduces the following changes: a backup node for the primary and no global failures on the switch interface. With the SP Switch, there is no problem shutting down a node that is currently on the Switch without explicitly fencing it because the Worm daemon of the primary node does this for you. As soon as you shutdown or power off a node, this node get fenced.

This is also true for the primary node: the primary backup node immediately becomes the primary node, and after finishing the recovery procedure, the old primary node is set to *isolated* in the SDR.

10.2.8 Modifying the Switch Values in the SDR

There are two situations where you are forced to modify the SDR manually. Assume the worst case: all nodes are on isolated in the SDR, or a few nodes are isolated and you want to have one of these nodes as primary. In this case you will not be able to start the Switch. But as long as the Switch is not up, the isolated entry in the SDR is just a value that can be changed manually because these entries will not be evaluated before `Estart`.

By the way*...what criteria do you use to say the Switch is up and running?* It is up as soon as the primary and primary backup nodes have initialized the switch interface. After that, all the other nodes can be integrated by using the `Eunfence` command without any `Estart`.

Before changing the SDR manually, make a backup by using the following command:

```
# SDRArchive
```

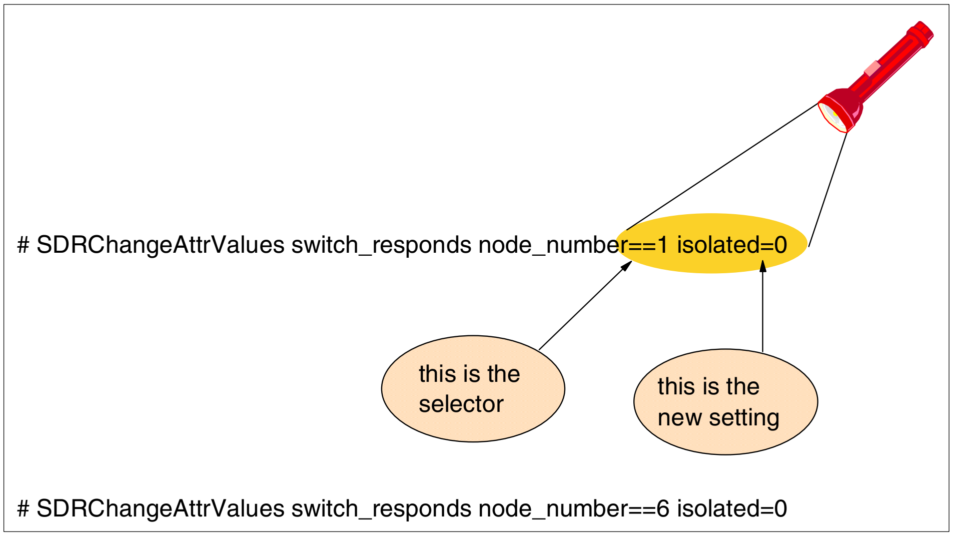
This command will create a tar-file of the SDR in the subdirectory /spdata/sys1/install/archives (refer to 7.1, “Backing Up Your SP System” on page 139). The following screen shows an example output from the SDRArchive command.

```
[sp4en0:/]# SDRArchive
SDRArchive: SDR archive file name is /spdata/sys1/sdr/archives/backup.99109.1320
[sp4en0:/]#
```

Check the Switch Responds by using the `SDRGetObjects switch_responds` command. The next screen shows example output from the command:

```
# SDRGetObjects switch_responds
node_number  switch_responds autojoin     isolated     adapter_config_status
           1            0            0            1 css_ready
           5            0            0            1 css_ready
           6            0            0            1 css_ready
           7            0            0            1 css_ready
           8            0            0            1 css_ready
           9            0            0            1 css_ready
          10            0            0            1 css_ready
          11            0            0            1 css_ready
          12            0            0            1 css_ready
          13            0            0            1 css_ready
          14            0            0            1 css_ready
          15            0            0            1 css_ready
```

To get the Switch up, at least the isolated value of primary and primary backup must be 0 in the SDR. The primary and primary backup are nodes 1 and 6. Figure 48 on page 237 shows the two commands for changing the isolated values of these nodes.



```
# SDRChangeAttrValues switch_responds node_number==1 isolated=0
```

```
# SDRChangeAttrValues switch_responds node_number==6 isolated=0
```

Figure 48. SDRChangeAttrValues

After setting these new values, `SDRGetObjects switch_responds` looks like the following:

```
# SDRGetObjects switch_responds
node_number  switch_responds autojoin     isolated     adapter_config_status
           1            0            0            0 css_ready
           5            0            0            1 css_ready
           6            0            0            0 css_ready
           7            0            0            1 css_ready
           8            0            0            1 css_ready
           9            0            0            1 css_ready
          10            0            0            1 css_ready
          11            0            0            1 css_ready
          12            0            0            1 css_ready
          13            0            0            1 css_ready
          14            0            0            1 css_ready
          15            0            0            1 css_ready
```

If Kerberos is working and the Worm daemon is running on nodes 1 and 6, `Estart` will succeed for these two nodes.

```
# SDRChangeAttrValues switch_responds isolated=0
```

The preceding command will change every isolated attribute value found in the SDR class *switch_responds*, as follows:

```
# SDRChangeAttrValues switch_responds isolated=0
node_number  switch_responds autojoin     isolated     adapter_config_status
          1            0            0            0 css_ready
          5            0            0            0 css_ready
          6            0            0            0 css_ready
          7            0            0            0 css_ready
          8            0            0            0 css_ready
          9            0            0            0 css_ready
         10            0            0            0 css_ready
         11            0            0            0 css_ready
         12            0            0            0 css_ready
         13            0            0            0 css_ready
         14            0            0            0 css_ready
         15            0            0            0 css_ready
```

The reason for using the `SDRArchive` before using this `SDRChangeAttrValues` command is obvious: if you look at the command syntax for the SDRChangeAttrValues command, you will notice that the node gets selected by using a `node_number==6`. Using the double equal sign (==) is very important; if you use a single equal sign (=) instead, *all* the isolated values of all nodes will be set to the new value. If this happens, use your SDR archive and restore it. Here is an example how to restore the SDR:

```
# SDRRestore <backup filename>
```

For more information on this subject, refer to 7.2.2, “Restoring the SDR” on page 149.

10.3 Monitoring the Switch

With PSSP 2.3, the Emonitor daemon has been introduced. This daemon, which is managed by the system resource controller, only runs on the CWS. It is not started automatically; therefore, you have to edit the appropriate configuration file and then start the Switch with the `Estart -m` command. Currently the switch is up and running on our system and started without the monitor option. Let us check the SP-related subsystems, as follows:

```
# lssrc -a | grep sp4en0
hags.sp4en0       hags           18834   active
hagsglsm.sp4en0   hags           19608   active
haem.sp4en0       haem           21156   active
haemaixos.sp4en0  haem           20902   active
hats.sp4en0       hats           20134   active
hr.sp4en0         hr             21418   active
pman.sp4en0       pman           21674   active
pmanrm.sp4en0     pman           20648   active
sdr.sp4en0        sdr            11360   active
Emonitor.sp4en0   emon                   inoperative
```

As we see, the Emonitor.sp4en0 subsystem is inoperative.

10.3.1 How to Activate the Emonitor Function

First you have to edit the /etc/SP/Emonitor.cfg file to indicate which nodes should be monitored. You specify these nodes at the end of the file. In the following screen, we are looking at the end of this file:

```
# vi /etc/SP/Emonitor.cfg
<Shift G>
# example entries (remove # to use
#5  # This will set node 5 to be monitored
#1  # This will set node 1 to be monitored
1
5
6
7
8
9
10
```

We added nodes 1, 5, 6, 7, 8, 9 and 10 to this file. Next we have to start the Switch with the monitor option:

```
# Estart -m
```

As we see, the Emonitor subsystem is now activated.

```
# lssrc -a | grep emon
Emonitor.sp4en0   emon           23556   active
```

10.3.2 How Emonitor Works

The `Estart -m` command starts the Switch in monitoring mode. Contrary to what you might expect, Emonitor checks the nodes' host_responds values in the SDR, not the switch_reponds. As soon as the host_responds of a node turns to 0 (red), the Emonitor daemon detects it and when the host_responds for this node returns, it integrates this node by the `Eunfence` command (see Figure 49 for this sequence).

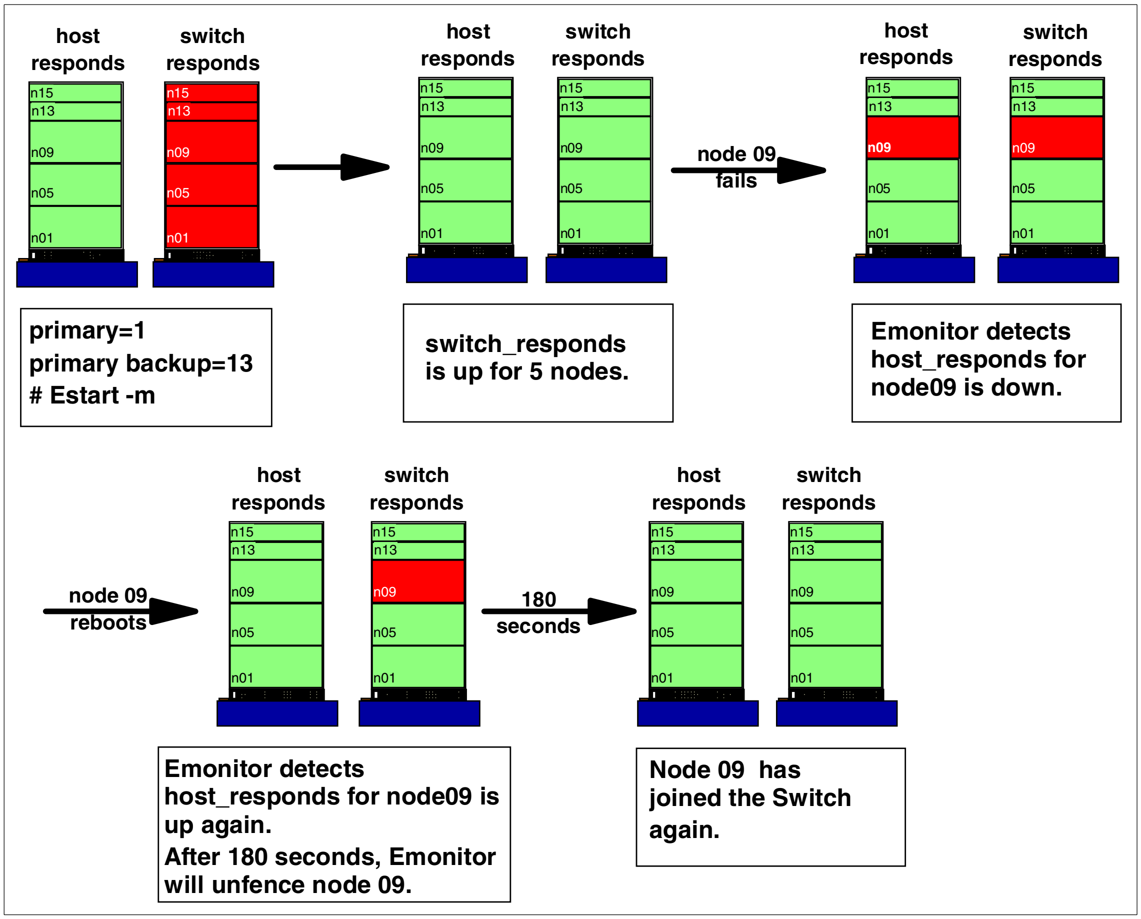
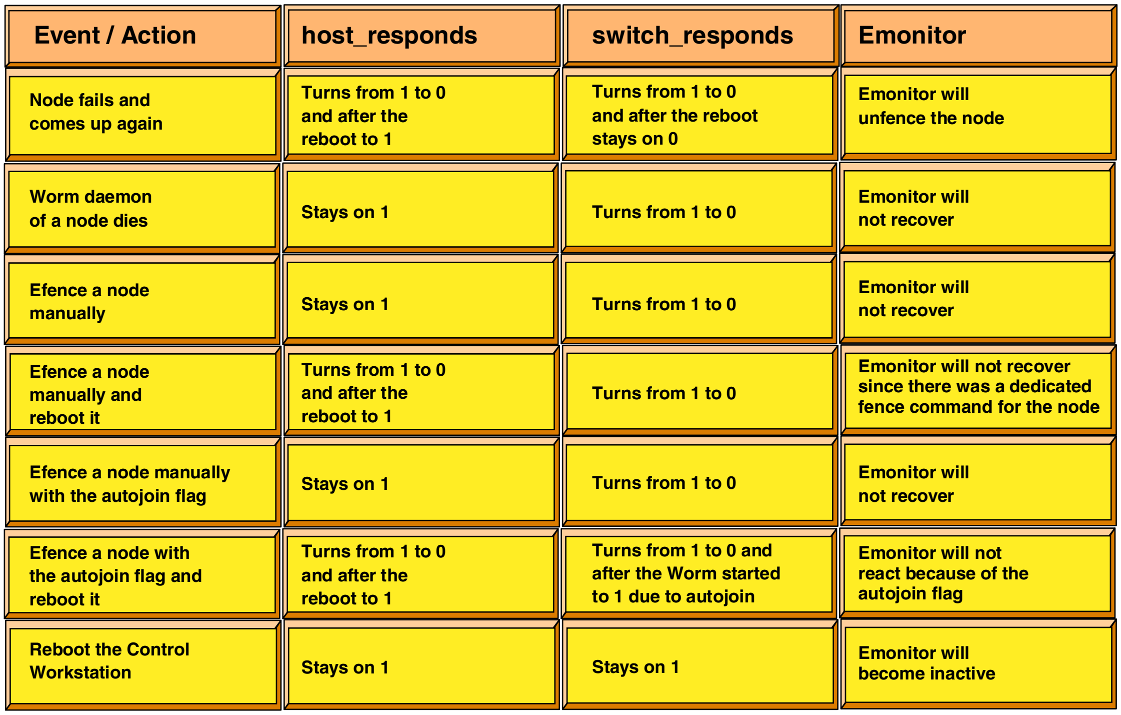

host responds
switch responds
n15
n13
n09
n05
n01
primary=1
primary backup=13
# Estart -m
switch_responds is up for 5 nodes.
node 09 fails
Emonitor detects host_responds for node09 is down.
node 09 reboots
Emonitor detects host_responds for node09 is up again.
After 180 seconds, Emonitor will unfence node 09.
180 seconds
Node 09 has joined the Switch again.

Figure 49. Emonitor

That means Emonitor will not recover from a “death” of a node’s Worm daemon because there is no impact on the host_responds value of this node. Furthermore, Emonitor is not able to restart the Worm daemon.

If a node is under control of Emonitor and this node crashes, when it comes up again, first the `rc.switch` script starts the Worm daemon. As soon as the host_responds for this node turns to 1 (green), Emonitor detects it, waits for a time interval of 180 seconds, then checks in the SDR if the autojoin flag is set for this node. If yes, Emonitor does nothing because autojoin will be covered from the Worm daemon. If no, Emonitor executes `Eunfence` for this node and reintegrates it to the Switch. Figure 50 shows the cases and the recovery actions.



Event / Action	host_responds	switch_responds	Emonitor
Node fails and comes up again	Turns from 1 to 0 and after the reboot to 1	Turns from 1 to 0 and after the reboot stays on 0	Emonitor will unfence the node
Worm daemon of a node dies	Stays on 1	Turns from 1 to 0	Emonitor will not recover
Efence a node manually	Stays on 1	Turns from 1 to 0	Emonitor will not recover
Efence a node manually and reboot it	Turns from 1 to 0 and after the reboot to 1	Turns from 1 to 0	Emonitor will not recover since there was a dedicated fence command for the node
Efence a node manually with the autojoin flag	Stays on 1	Turns from 1 to 0	Emonitor will not recover
Efence a node with the autojoin flag and reboot it	Turns from 1 to 0 and after the reboot to 1	Turns from 1 to 0 and after the Worm started to 1 due to autojoin	Emonitor will not react because of the autojoin flag
Reboot the Control Workstation	Stays on 1	Stays on 1	Emonitor will become inactive

Figure 50. Emonitor Cases

10.3.3 The Emonitor.log File

When you start the Switch with the monitor flag, the Emonitor.log file is created under /var/adm/SPlogs/css. This file provides you with the information about the events that have been detected by the Emonitor daemon and also the decisions about the action. In the following example, we fenced node 7 manually and rebooted it. Due to our `Efence` command, Emonitor decided not to unfence it when the host_responds was back.

```
# cat /var/adm/SPlogs/css/Emonitor.log
Emonitor: Monitoring host_responds Wed Apr 14 09:44:00 EDT 1999 partition sp4en0
.
Emonitor: frame1 node10 came up. Wed Apr 14 09:49:36 EDT 1999 partition sp4en0
Emonitor: Timer popped .... Wed Apr 14 09:50:16 EDT 1999
 partition sp4en0
Emonitor: Looking For Nodes: partition sp4en0
 1[ OK ]
 5[ OK ]
 6[ OK ]
 7[Autounfence in effect .. unfence NOT required]
 8[ OK ]
 9[ OK ]
 10[ OK ]
 11[ OK ]
 12[ OK ]
 13[ OK ]
 14[ OK ]
 15[ OK ]

Emonitor: Timer action completed partition sp4en0.
```

10.3.4 The Time Interval

Since `Estart -m` does not permit any other options, you cannot manipulate the time interval between the detection of returning host_responds of a node and the `Eunfence` command issued by Emonitor. The time interval is 180 seconds.

While we do not recommend it, if you think this time period is too long for your system, you can change this value in the /usr/lpp/ssp/bin/Emonitor file by editing the entry $EstartSTALL=180 to $EstartSTALL=<seconds>. Bear in mind that this could be changed again after applying PSSP PTFs.

10.3.5 Monitoring the Switch with PSSP 3.1

With PSSP 3.1, the *cssadm* daemon has been introduced for monitoring the Switch. This daemon is managed by the system resource controller and gets its information from Event Management. For further information about using this daemon, refer to *Understanding and Using the SP Switch,* SG24-5161.

Appendix A. Downloading PTFs for RS/6000 and SP

IBM provides a number of mirrored sites on the Internet where you may freely download AIX-related fixes. While not every AIX-related fix is available, we are constantly adding to these anonymous FTP servers. Though we do not guarantee all fixes will be immediately made available, we usually update the servers within 24 hours of tape distribution.

The current anonymous FTP servers are:

Country	Hostname	IP Address
United States	service.boulder.ibm.com	198.17.57.66
United Kingdom	ftp.europe.ibm.com	193.129.186.2
Canada	rwww.aix.can.ibm.com	204.138.188.126
Germany	www.ibm.de	192.109.81.2
Japan	fixdist.yamato.ibm.co.jp	203.141.89.41

To download the fixes on these servers you can use either the `ftp` command, or a Web browser, or the AIX-exclusive tool called FixDist.

This appendix discusses downloading fixes using the FixDist tool and from the Web.

A.1 FixDist Tool

The FixDist tool is a Web application that provides discrete downloads and delivers all required images with just one click. It can also keep track of fixes you have already downloaded, so you can download smaller fix packages the next time you need them.

Although the location of the Web pages varies from country to country, the most common on is:

`http://service.boulder.ibm.com/support/rs6000`

The FixDist tool and the user's guide are located in the anonymous FTP directory /aix/tools/fixdist, or they can be viewed online with a Web browser.

A.1.1 Downloading the FixDist Tool

To download through the Web:

1. Go to the AIX Technical Support home page.

   `http://service.boulder.ibm.com/support/rs6000`

2. Click **Downloads**.
3. Click **Tools**.
4. Download the tool and guide into your /tmp directory by clicking the appropriate links.

To download using the `ftp` command:

1. Change to your /tmp directory.
2. Get the tool and the PostScript user's guide. The text version of the user's guide comes with the tool and once installed is found as /usr/lpp/fixdist/fixdist.txt.

You can use any hostname from the list in the previous Electronic Fix Distribution section:

```
# ftp service.software.ibm.com
> login:     anonymous
> password:  "email"   (example: johndoe@)
> bin
> cd /aix/tools/fixdist
> get fd.tar.Z         (FixDist tool in compressed tar format)
> get fixdist.ps.Z     (User guide in compressed PostScript)
> quit
```

A.1.2 Installing the FixDist Tool

The following procedure must be performed as the root user.

Install the tool into the /usr file system. You *must* install it from the / (root) directory to access the online help and preserve your .netrc file.

```
# cd /                             (change directory to the root)

# zcat /tmp/fd.tar.Z |tar -xpvf -   (uncompress and untar)
```

A.1.3 Starting and Configuring the FixDist Tool

The database and temporary files that FixDist manipulates on your RS/6000 are very sensitive to file permissions. Because of this, we recommend that you consistently run FixDist using the same user ID you start with.

To start FixDist, enter:

```
# fixdist
```

The file /usr/bin/fixdist is a script that calls /usr/lpp/fifdist/fixdistm for CDE and AIXwindows users. If you have a dumb terminal, FixDist will call /usr/lpp/fixdist/fixdistc.

Read the user's guide for detailed configuration information. The basic configuration tasks are: specifying an IBM server; specifying a location on your RS/6000 where you want to put the fixes; downloading the fix database from the IBM server to your RS/6000.

A.2 TapeGen Tool

TapeGen is a companion service tool provided with FixDist that enables you to create a stacked tape containing SMIT-installable fixes. You create a stack file that lists all the images you want stacked onto a tape and TapeGen does the rest.

A.3 Downloading Fixes from the Web

This section contains the steps for downloading fixes from the Web. It is applicable to AIX Versions 3 and 4.

The steps for downloading are as follows:

1. Open the following URL:

   `http://service.software.ibm.com/aix.us/`

2. Select the database to use (AIX Version 3/AIX Version 4/CATIA for AIX).
3. Select the search options (APAR NUMBER/FILESET NAME/PTF NUMBER/APAR ABSTRACT).
4. Type in the fix number desired (APAR NUMBER/PTF NUMBER) and click **submit**.
5. Use your left mouse button to click on the needed updates.
6. Select your Maintenance level.

7. Click [**Get Fix Package**].
8. Use your right mouse button to click the image needed and select the option to **Save Link as ... .**
9. Select the directory to save the file and use your left mouse button to click [**Save**].

Appendix B. Integrating an S70 System into an SP

PSSP version 3.1 provides support for integrating the S70 system as a node to SP. In this appendix we discuss how to integrate S70 to SP.

B.1 Overview

The S70 node in the SP has the following characteristics:

- The node is not physically located in the SP frame. It will be considered as a non-SP frame node. This frame will be assigned a frame number containing a single node. Therefore, a S70 frame will occupy 16 node numbers, of which only one will ever be used.
- The node is connected to the SP administrative network (also referred to as the SP LAN).
- The node is attached to the switch with a TB3PCI adapter.
- There is no frame or node supervisor card. The S70 node will be connected to the SP Control Workstation using two serial ports. One tty port is used for the hardware_protocol, and the second tty port is for establishing the serial connection using the `s1term` command.
- The S70 node will contain all the PSSP code, and be managed and used in all the same ways as standard SP nodes currently are.
- The S70 frame cannot be the first frame; an SP frame must always be the lowest numbered frame.

B.2 Installation

Installation of the S70 node in SP is the same as any other node, after you set up the physical connections between the CWS and the S70 system. As far as the installation steps are concerned, the only difference you will find is when configuring the frame information. For incorporating the S70 node (which, as previously mentioned, is considered as a non-SP frame), the Enter Database Information SMIT panel has a new option; the Non-SP Frame Information menu. This menu has to be used for configuring the S70 frames.

Lab Scenario:

We have a single frame SP system with four High nodes and a Control Workstation. In this lab, we will integrate the S70 system to our existing SP environment. Figure 51 on page 248 shows the configuration of the SP and the S70 node.

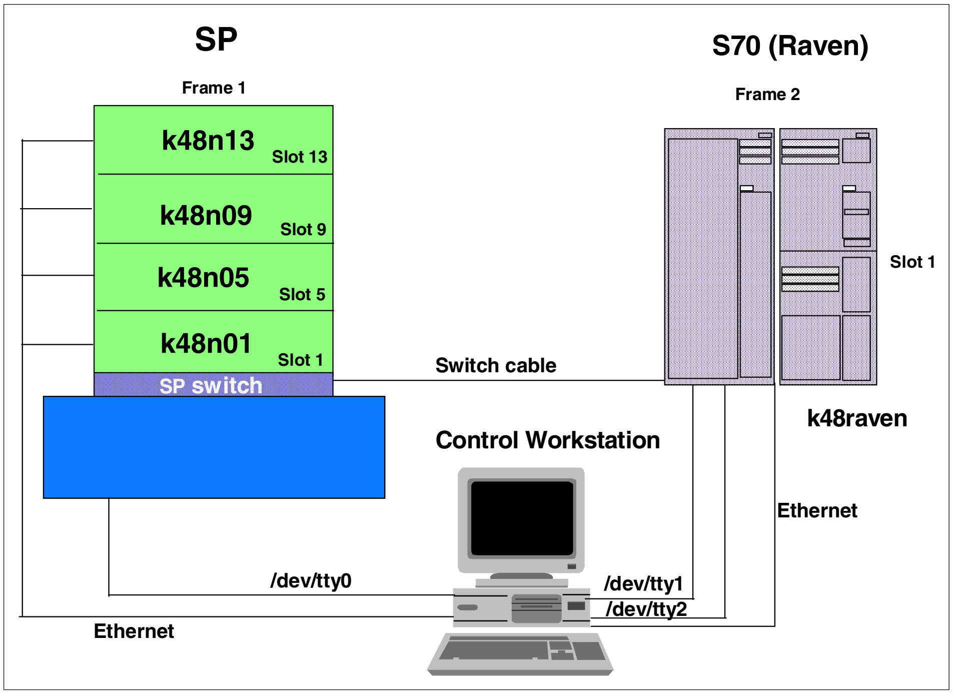


Figure 51. Lab Scenario of Integrating S70 to an SP

The steps to integrate the S70 system to the existing SP are as follows:

Step 1: Connect the S70 system to the Control Workstation

- Connect two RS232 cables from the tty ports of the S70 system to the SP Control Workstation on ports tty1 and tty2.
- Connect the Ethernet port to the SP LAN.
- If you have the switch adapter TB3PCI present on the S70 system, connect it to the SP switch.

Step 2: Power on the S70 system

If the S70 is not already powered on, turn it on.

Step 3: Configure the RS-232 Control line

Configure the tty ports /dev/tty1 and /dev/tty2, which are connected to the S70 system from the CWS by using the `mkdev` command as follows:

This example configures the second tty port in the CWS.

```
# mkdev -c tty -t tty -s rs232 -p sa1 -w s2
```

This example configures port 0 of the 16-Port Asynchronous Adapter EIA-232:

```
# mkdev -c tty -t tty -s rs232 -p sa2 -w 0
```

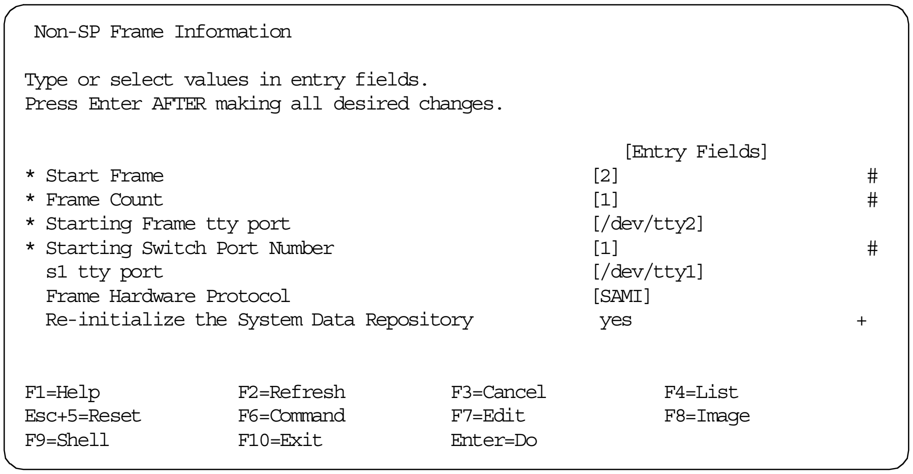
Step 4: Enter frame information and initialize the SDR

In our lab scenario, the S70 system will be the second frame. Port tty1 will be used for the s1term, and port tty2 will be used for the hardware protocol. The hardware_protocol for S70 nodes is Service and Manufacturing Interface (SAMI).

- Add the frame definition of the S70 node in the SP as follows:

  ```
  # smitty enter_data
  ```

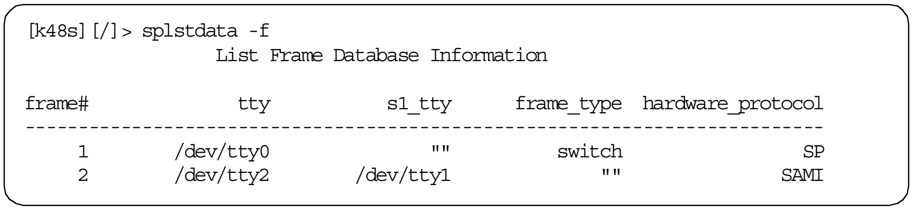
  Select **Non-SP Frame Information.**



```
 Non-SP Frame Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Start Frame                                    [2]                 #
* Frame Count                                    [1]                 #
* Starting Frame tty port                        [/dev/tty2]
* Starting Switch Port Number                    [1]                 #
  s1 tty port                                    [/dev/tty1]
  Frame Hardware Protocol                        [SAMI]
  Re-initialize the System Data Repository        yes               +


F1=Help            F2=Refresh         F3=Cancel          F4=List
Esc+5=Reset        F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

Step 5: Verify frame information

Check the frame configuration by using the command `splstdata -f` to verify that the S70 system is added as frame 2. The output of the command will look as follows:



```
[k48s][/]> splstdata -f
                  List Frame Database Information

frame#             tty          s1_tty      frame_type  hardware_protocol
----------------------------------------------------------------------------
     1       /dev/tty0              ""          switch                 SP
     2       /dev/tty2       /dev/tty1              ""               SAMI
```

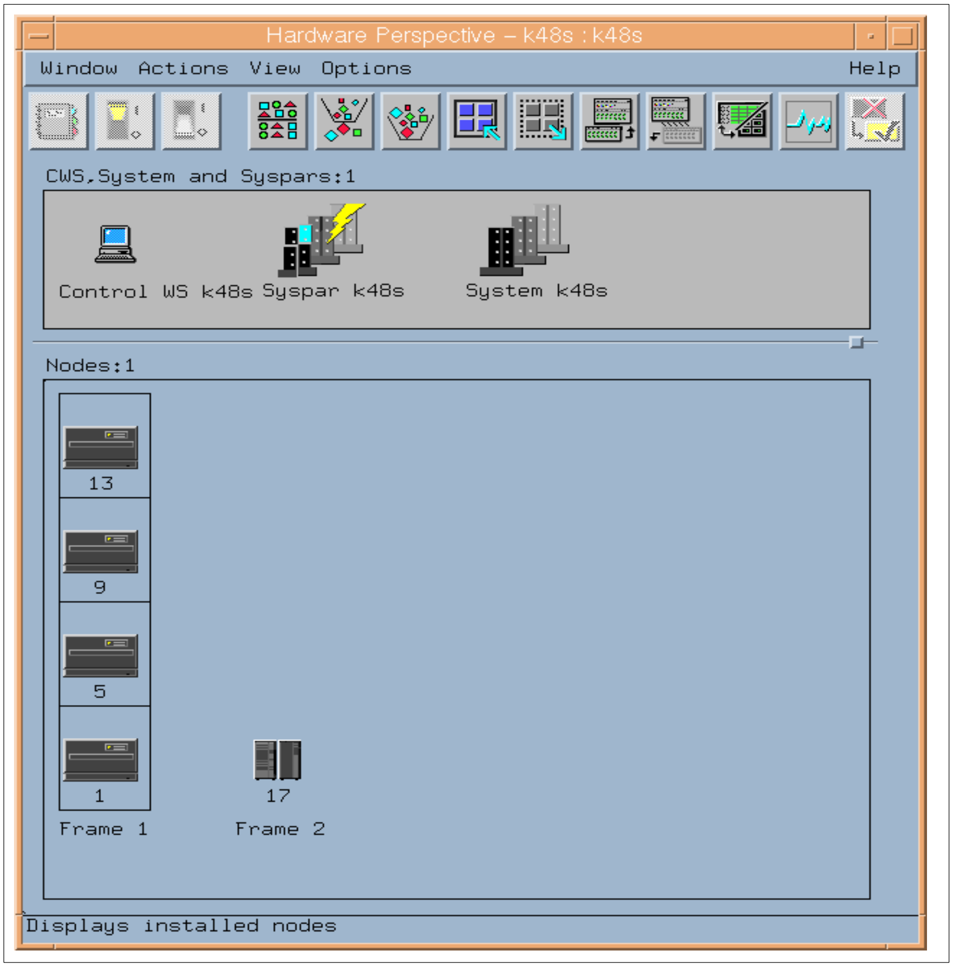
Step 6: Node object population

Since there is no node or frame supervisor card on the S70 node, there will be no supervisor-to-supervisor communication to tell hardmon to trigger a logging event. For S70 nodes, population of the node object will be triggered by the `spframe` command executed in the CWS. The S70 node does not need any microcode download as it does not have any frame or node supervisor cards.

When we update the frame definition in the SDR, the node k48raven.ppd.pok is configured in frame 2, slot 1 and the node number is 17. Check this using the command `splstdata -n`. The screen output will look as follows:

```
k48s][/]> splstdata -n
                 List Node Configuration Information

node# frame# slot# slots  initial_hostname  reliable_hostname  dcehostname
      default_route   processor_type processors_installed description
-----------------------------------------------------------------------------
    1      1     1     4  k48n01.ppd.pok.i   k48n01.ppd.pok.i  ""
          9.114.88.93            MP                  1 ""
    5      1     5     4  k48n05.ppd.pok.i   k48n05.ppd.pok.i  ""
          9.114.88.93            MP                  1 ""
    9      1     9     4  k48n09.ppd.pok.i   k48n09.ppd.pok.i  ""
          9.114.88.93            MP                  1 ""
   13      1    13     4  k48n013.ppd.pok.   k48n013.ppd.pok.  ""
          9.114.88.93            MP                  1 ""
   17      2     1     1  k48raven.ppd.pok   k48raven.ppd.pok  ""
          9.114.88.93            MP                  1 ""
[k48s][/]>
```

The frame and node information in Perspectives will look as shown in Figure 52 on page 251.

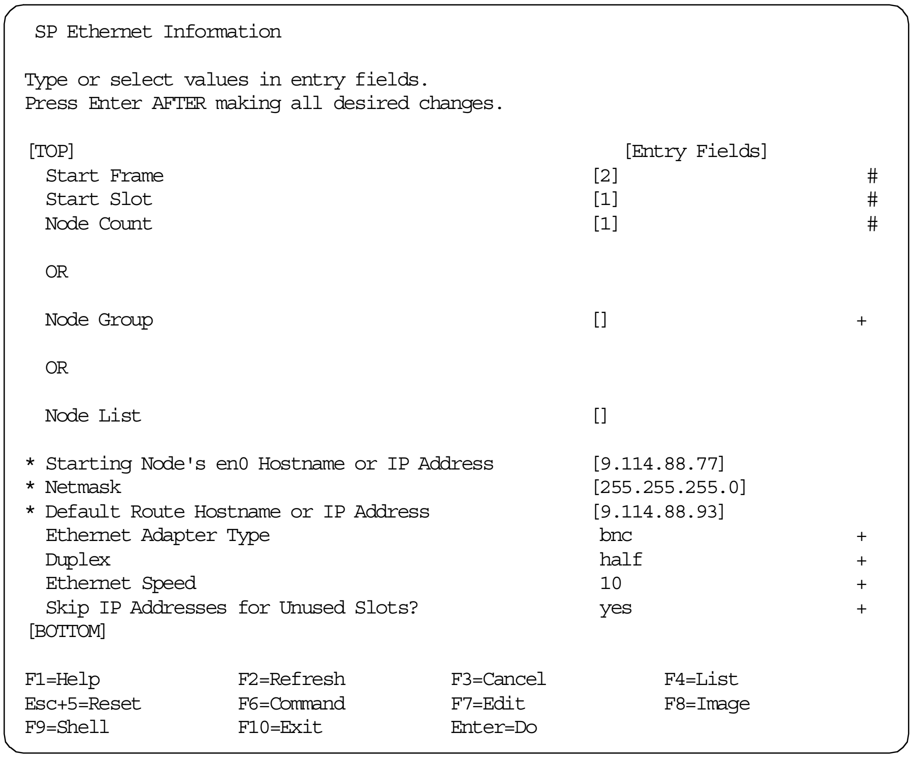


Displays installed nodes

Figure 52. Frame/Node Information in Perspectives

Step 7: Enter the required node information:

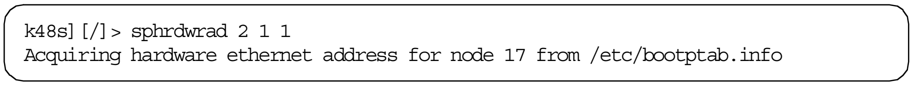
This step adds IP address-related information to the node object in the SDR. This information is used for node customization and configuration. Configure the S70 node Ethernet information and the default route as follows:



```
 SP Ethernet Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
  Start Frame                                       [2]                    #
  Start Slot                                        [1]                    #
  Node Count                                        [1]                    #

  OR

  Node Group                                        []                    +

  OR

  Node List                                         []

* Starting Node's en0 Hostname or IP Address        [9.114.88.77]
* Netmask                                           [255.255.255.0]
* Default Route Hostname or IP Address              [9.114.88.93]
  Ethernet Adapter Type                              bnc                  +
  Duplex                                             half                 +
  Ethernet Speed                                     10                   +
  Skip IP Addresses for Unused Slots?                yes                  +
[BOTTOM]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Step 8: Acquire the Hardware Ethernet address

The next step is to acquire the hardware Ethernet address of the Ethernet adapter in the S70 system using the command `sphrdwrad`. If you already know the hardware MAC address of the Ethernet adapter in the S70 system, put the address in the file /etc/bootptab.info to speed up the process. Keep in mind that to get the hardware Ethernet address, the node will be powered off and on. This example gets the hardware Ethernet address for the S70 node in frame 2, slot 1 and node count 1.



```
k48s][/]> sphrdwrad 2 1 1
Acquiring hardware ethernet address for node 17 from /etc/bootptab.info
```

Step 9: Verify that the Ethernet addresses were acquired

Check that the Ethernet hardware addresses are placed in the SDR node object by using the following command:

```
[k48s][/]> splstdata -b -l 17
                List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response           install_disk
     last_install_image   last_install_time  next_install_image lppsource_name
              pssp_ver         selected_vg
------------------------------------------------------------------------------
   17 k48raven.ppd.pok   020701232D9C    0      install               hdisk0
               initial            initial     bos.obj.ssp.432           aix432
              PSSP-3.1             rootvg
[k48s][/]>
```

Step 10: Configure additional adapters in the S70 node

Depending on the system configuration you have in the S70 system, configure all the network and switch adapters now by using the command `spadaptrs` or by using SMIT/Perspectives.

Step 11: Configure the initial hostname for the S70 node

Configure the initial hostname for the S70 node using the command `sphostnam`. This command indicates that the hostname is the fully qualified form of the hostname for the en0 adapter, for the frame 2, start slot 1 and node count of 1.

```
# sphostnam -a en0 -f long 2 1 1
```

Step 12: Refresh system partition-sensitive subsystems

To refresh the subsystems, issue the following command on the Control Workstation:

```
# syspar_ctrl -r -G
```

This command refreshes all the subsystems in each partition.

Step 13: Install the system image in the S70 node

Select the image and PSSP version to be installed by using the `spbootins and spchvgobj` commands, or by using SMIT/Perspectives.

For example, to install the image bos.obj.ssp.432 and PSSP-3.1 on the S70 node, use the following commands:

```
[k48s][/]> spchvgobj -r rootvg -i bos.obj.ssp.432 -p PSSP-3.1 -v aix432 2 1 1
spchvgobj: Successfully changed the Node and Volume_Group objects for node number
volume group rootvg.
spchvgobj: The total number of changes successfully completed is 1.
spchvgobj: The total number of changes which were not successfully completed is 0
[k48s][/]>
[k48s][/]> spbootins -s no -r install 2 1 1
[k48s][/]>
[k48s][/]> setup_server
```

Step 14: Network-boot the S70 node

First open the LED/LCD display from Perspectives to monitor the installation codes. The LCD display screen has been changed to show the Raven node. This is shown in Figure 53.

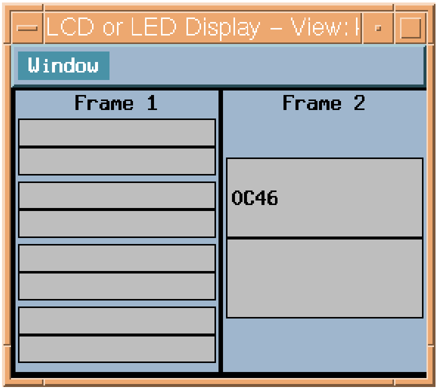


Figure 53. LED/LCD Display for the New S70 Node

To monitor the messages displayed during the time of installation, start the tty console for the S70 node by using the command:

```
# s1term 2 1
```

Now initiate the network boot on the S70 node and monitor the tty and LCD display to verify that the AIX and PSSP software is getting installed in the node.

Appendix C. Secondary Authentication Server on an SP Node

As mentioned in 8.12, “Setting Up a Secondary Authentication Server” on page 197, an SP node is not designated to become a secondary authentication server for the SP kerberos realm. But in case you want to have a secondary server and you are forced to install it on one of the SP nodes, this appendix provides the necessary workaround.

C.1 On the Appropriate Node

You have to first run `setup_authent` on the appropriate node. This command checks the node_number value in the CuAt ODM. If this value is 0 (for a Control Workstation), or if it does not exist at all, the setup of the secondary server will be started. Otherwise, the `setup_authent` script exits. The steps described in the following sections then have to be done.

C.2 On the Control Workstation

- Edit the /etc/krb.conf file on the primary authentication server and distribute it.

C.3 On the Node That Will Become the Secondary Authentication Server

- Make a copy of CuAt ODM.
- Write the node_number stanza of the ODM to a file.
- Delete the node_number stanza in ODM.
- Run `setup_authent.`
- Add the node_number stanza back to the ODM.

C.4 On the Control Workstation

Add an entry to the /etc/krb.conf file for the secondary server as described in 8.12.2, “/etc/krb.conf File Modification and Distribution” on page 199. Then distribute this modified /etc/krb.conf to all nodes belonging to the realm:

```
# pcp '-a' /etc/krb.conf
```

C.5 On the Future Secondary Authentication Server

Just to have a backup, make a copy of the CuAt ODM:

```
# cp /etc/objrepos/CuAt /etc/objrepos/CuAt.bak
```

```
# odmget -q name=sp CuAt > /tmp/node_num_stanza
```

```
# odmdelete -o CuAt name=sp
```

Now set up the secondary server with the `setup_authent` command (see also 8.12.3, “Run setup_authent on the Secondary Authentication Server” on page 200 for more information).

```
# /usr/lpp/ssp/bin/setup_authent
```

Reintegrate the node_number stanza to the ODM:

```
# odmadd /tmp/node_num_stanza
```

The secondary server setup is now complete.

Appendix D. Special Notices

This publication is intended to help RS/6000 SP system administrators, system operators and field personnel to maintain the RS/6000 SP. It provides techniques for installing, migrating and maintaining an RS/6000 SP environment. The information in this publication is not intended as the specification of any programming interfaces that are provided by RS/6000 SP. See the PUBLICATIONS section of the IBM Programming Announcement for RS/6000 SP 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, North Castle Drive, Armonk, NY 10504-1785.

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 information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. 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.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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 ®	AT
RS/6000	SP

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries. (For a complete list of Intel trademarks see www.intel.com/tradmarx.htm)

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

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 263.

- *PSSP 2.4 Technical Presentation,* SG24-5173
- *PSSP 3.1 Announcement,* SG24-5332
- *RS/6000 SP: Problem Determination Guide,* SG24-4778
- *RS/6000 SP: PSSP 2.2 Survival Guide,* SG24-4928
- *RS/6000 SP High Availability Infrastructure,* SG24-4838
- *Understanding and Using the SP Switch,* SG24-5161
- *Inside the RS/6000 SP*, SG24-5145

E.2 Redbooks on CD-ROMs

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at `http://www.redbooks.ibm.com/` for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	**Collection Kit Number**
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037

E.3 Other Publications

These publications are also relevant as further information sources:

- *IBM Parallel System Support Programs for AIX: Installation and Migration Guide,* GC23-3898
- *IBM Parallel System Support Programs for AIX: Administration Guide,* GC23-3897
- *IBM Parallel System Support Programs for AIX: Command and Technical Reference, Volume 1 and Volume 2,* SA22-7351
- *AIX Version 4.3, Network Installation Management Guide and Reference*, SC23-4113

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** `http://www.redbooks.ibm.com/`

  Search for, view, download, or order hardcopy/CD-ROM redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at `http://w3.itso.ibm.com/` and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access `MyNews` at `http://w3.ibm.com/` for redbook, residency, and workshop announcements.

IBM Redbook Fax Order Form

Please send me the following:

Title	**Order Number**	**Quantity**

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.

List of Abbreviations

AIX	Advanced Interactive Executive
AMG	Adapter Membership Group
ANS	Abstract Notation Syntax
APAR	authorized program analysis report
API	Application Programming Interface
BIS	boot/install server
BSD	Berkeley Software Distribution
BUMP	Bring-Up Microprocessor
CP	Crown Prince
CPU	central processing unit
CSS	communication subsystem
CWS	control workstation
DB	database
EM	Event Management
EMAPI	Event Management Application Programming Interface
EMCDB	Event Management Configuration Database
EMD	Event Manager Daemon
EPROM	Erasable Programmable Read-Only Memory
FIFO	first-in first-out
FS	file system
GB	gigabytes
GL	Group Leader
GPFS	General Parallel File System
GS	Group Services
GSAPI	Group Services Application Programming Interface
GVG	Global Volume Group
HACMP	High Availability Cluster Multiprocessing
HACMP/ES	High Availability Cluster Multiprocessing Enhanced Scalability
hb	heart beat
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	Journaled File System
LAN	Local Area Network
LCD	liquid crystal display
LED	light emitter diode
LP	logical partition
LPP	Licensed Program Product
LRU	last recently used
LSC	Link Switch Chip
LV	logical volume

LVM	Logical Volume Manager
MB	megabytes
MIB	Management Information Base
ML	Maintenance Level
MPI	Message Passing Interface
MPL	Message Passing Library
MPP	Massive Parallel Processors
NFS	Network File System
NIM	Network Installation Management
NSB	Node Switch Board
NSC	Node Switch Chip
OID	object ID
ODM	Object Data Manager
PAIDE	Performance Aide for AIX
PE	Parallel Environment
PID	process ID
POE	Parallel Operating Environment
PP	physical partition
PSSP	Parallel System Support Programs
PTC	prepare to commit
PTF	Program Temporary Fix
PTPE	Performance Toolbox Parallel Extensions
PTX	Performance Toolbox for AIX
PV	physical volume
RAM	random access memory
RCP	Remote Copy Protocol
RML	Recommended Maintenance Level
RM	Resource Monitor
RMAPI	Resource Monitor Application Programming Interface
RPQ	Request For Product Quotation
RSCT	RS/6000 Cluster Technology
RSI	Remote Statistics Interface
R/VSD	Recoverable/Virtual Shared Disk
RVSD	Recoverable Virtual Shared Disk
SAMI	Service and Manufacturing Interface
SBS	structured byte string
SCSI	Small Computer System Interface
SDR	System Data Repository
SMIT	System Management Interface Tool
SSA	Serial Storage Architecture
VG	volume group
VRMF	version, release, modification, and fix
VSD	Virtual Shared Disk
WEBSM	Web-based System Manager

Index

ITSO Redbook Evaluation

RS/6000 SP Software Maintenance
SG24-5160-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to `redbook@us.ibm.com`

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction __________

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

____

____

____

____

What other redbooks would you like to see published?

____

____

____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

____

____

____

____

____

SG24-5160-00

Printed in the U.S.A.

RS/6000 SP Software Maintenance

SG24-5160-00