AIX 5L Version 5.1

Network Installation Management Guide and Reference

AIX 5L Version 5.1

Network Installation Management Guide and Reference

Fifth Edition (April 2001)

Before using the information in this book, read the general information in “Appendix E. Notices” on page 171.

This edition applies to AIX 5L Version 5.1 and to all subsequent releases of this product until otherwise indicated in new editions.

A reader’s comment form is provided at the back of this publication. If the form has been removed, address comments to Publications Department, Internal Zip 9561, 11400 Burnet Road, Austin, Texas 78758-3493. To send comments electronically, use this commercial Internet address: aix6kpub@austin.ibm.com. Any information that you supply may be used without incurring any obligation to you.

Contents

About This Book

AIX 5L Version 5.1 Network Installation Management Guide and Reference provides information about managing the installation and configuration of software by using a network interface. Network Installation Management (NIM) enables you to centrally manage the installation of the Base Operating System (BOS) and optional software on machines within a networked environment. NIM enables you to install the operating system on one or more machines from a server that already has AIX installed.

This guide does not:

- Describe how to install BOS onto a single machine without the use of a network installation server. For complete information about installing BOS, see *AIX 5L Version 5.1 Installation Guide*.
- Describe how to manage networks. See *AIX 5L Version 5.1 System Management Guide: Communications and Networks* for information about network management.

Who Should Use This Book

This guide is for system administrators who manage the installation and configuration of one or more AIX machines in a network environment. Readers should be familiar with the installation procedures in *AIX 5L Version 5.1 Installation Guide*, understand AIX system administration, and know how to manage a network environment.

Highlighting

The following highlighting conventions are used in this book:

Bold	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
Italics	Identifies parameters whose actual names or values are to be supplied by the user.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related Publications

The following books supplement the information in *AIX 5L Version 5.1 Network Installation Management Guide and Reference*:

- *AIX 5L Version 5.1 Installation Guide*
- *AIX 5L Version 5.1 System Management Guide: Communications and Networks*
- *AIX 5L Version 5.1 Commands Reference*
- *AIX 5L Version 5.1 System Management Guide: Operating System and Devices*

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- AIX
- IBM

Other company, product, or service names may be the trademarks or service marks of others.

Chapter 1. Network Installation Management (NIM) Introduction

This section provides an introduction to AIX Network Installation Management (NIM) and the operations you can perform to manage the installation of the Base Operating System (BOS) and optional software on one or more machines.

The types of machines you can manage are *standalone*, *diskless*, and *dataless*. A *standalone* machine is one that can boot (start up) by itself. Diskless and dataless systems cannot boot by themselves. They must use remote resources to boot. *Diskless* systems have no disk drive. *Dataless* systems have a local disk drive but they cannot boot from it. This book provides concepts and procedures for setting up the NIM environment, initiating the installation of standalone machines, and initializing resources for diskless and dataless machines.

Using NIM, you can install a group of machines with a common configuration or customize an installation for the specific needs of a given machine. The number of machines you can install simultaneously depends on the throughput of your network, the disk access throughput of the installation servers, and the platform type of your servers.

The NIM environment comprises client and server machines. A *server* provides resources (for example, files and programs required for installation) to another machine. A machine that is dependent on a server to provide resources is known as a *client*. In this guide and reference, any machine that receives NIM resources is a client, although the same machine can also be a server in the overall network environment.

All operations on clients in the NIM environment require one or more resources. NIM resource objects represent files and directories that are used to support some type of NIM operation. Because NIM resources are ordinary file system objects in the AIX operating system, most of them are provided to clients with standard Network File System (NFS) software. This means that many resources must reside locally on the servers providing these resources, because NFS can only export file system objects that are stored on local media in the machines from which they are exported.

Most installation tasks in the NIM environment are performed from one server, called the *master*. A set of installation tasks can also be performed from NIM clients. Once the network installation setup is complete, users of standalone clients can, from the client, install software that is available on NIM servers.

The machines you want to manage in the 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. Network objects and their attributes reflect the physical characteristics of the network environment. This information does not affect the running of a physical network but is used internally by NIM for configuration information.

Each object in the NIM environment has a unique name that you specify when the object is defined. The NIM name is independent of any of the physical characteristics of the object it identifies and is only used for NIM operations. The benefit of unique names is that an operation can be performed using the NIM name without having to specify which physical attribute should be used. NIM determines which object attributes to use. For example, to easily identify NIM clients, the host name of the system can be used as the NIM object name, but these names are independent of each other. When an operation is performed on a machine, the NIM name is used, and all other data for the machine (including the host name) is retrieved from the NIM database.

Overview of Network Installation Management (NIM) Operations

Network Installation Management operations include:

- “Installing BOS”
- “Customizing (Installing) the Software”
- “Maintaining the Software”
- “Configuring the Machine”
- “Booting Diagnostics”
- “Booting in Maintenance Mode”
- “Initializing Diskless and Dataless Clients” on page 3
- “Installing BOS on an Alternate Disk” on page 3

Note: For detailed information about the install process, see *AIX Version 4.3 Installation Guide*.

Installing BOS

NIM can be used as a utility for installing the AIX Base Operating System (BOS) on machines. NIM supports installation using any of the three BOS install types:

- Overwrite
- Preservation
- Migration

NIM also supports the use of **mksysb** images, **bosinst.data** files, and other resources to install machines.

Customizing (Installing) the Software

Installation of software packages and updates can be performed over the network using NIM. The NIM master can ″push install″ software onto a remote client, or a client can ″pull install″ software from a server.

Maintaining the Software

From the NIM master, you can deinstall software that resides on client machines. NIM also gives you the capability of committing and rejecting updates over the network.

Configuring the Machine

NIM provides the capability of running shell scripts on client machines for configuration after software has been installed.

Booting Diagnostics

Hardware diagnostics can be performed on NIM clients using a diagnostic boot image from a NIM server, rather than booting from a diagnostic tape or CD-ROM. Not only does this eliminate the need for diagnostic boot media, it eliminates the need to have diagnostics installed on the local disks of machines.

Booting in Maintenance Mode

For maintenance operations, you can boot a NIM client into maintenance mode from the boot image on a NIM server instead of using a bootable tape or CD-ROM.

Initializing Diskless and Dataless Clients

You can manage resources for diskless and dataless clients from the NIM master.

Installing BOS on an Alternate Disk

NIM can be used to clone the running of **rootvg** (root volume group) to an alternate disk, or install a **mksysb** image to an alternate disk.

Chapter 2. Choosing a NIM Master

There is only one NIM master for each NIM environment. The master is fundamental to all operations in the NIM environment. This machine must be installed and configured before you can perform any NIM operations. The master must be a standalone machine, running AIX AIX 4.2 (or later) with the NIM master fileset installed. The master manages the NIM configuration database and provides the central point of administration for the NIM environment.

If you set up more than one NIM environment, each comprised of a master and one or more clients on a network, these environments are independent of each other. No configuration information is shared between the different masters.

The **rsh** command is used to remotely execute commands on clients. To use the **rsh** command, the **$HOME/ .rhosts** file (where **$HOME** is typically ″/″ for root) on the client is configured automatically by NIM when the client is initialized so that the master has the permissions required to run commands on the client as root. Once a NIM client is running, a user with root authority on a client can disable the master's permissions to push commands onto the client. A user with root authority on a client can also enable the master's push permissions for the client. For more information about the **rsh** command and the **.rhosts** file, see *AIX 5L Version 5.1 Commands Reference* and *AIX 5L Version 5.1 Files Reference*. For information on how a client can control the master's push permissions, see "Controlling the Master or Client" on page 35.

Consider the following factors when choosing a machine that will serve as the NIM master:

- Before the NIM master can be used to manage the installation of other machines in the network, you need to install AIX 4.2 (or later) Base Operating System (BOS) on the machine you have selected to be the master. Follow the instructions in Installing the Base Operating System in the *AIX Version 4.3 Installation Guide*. You cannot use NIM to install BOS on the master.
- The machine must have a standalone configuration. It cannot be a diskless or dataless machine.
- The machine must be able to communicate with all the machines participating in the NIM environment.
- The machine and its console should be in a convenient location for you to perform network installation management tasks, because most operations are performed centrally from the master.
- The machine needs sufficient resources, such as disk space and system throughput, if you will use the master to serve NIM resources.

Chapter 3. Basic NIM Operations and Configuration

This section describes the following procedures for performing basic NIM operations and simple configuration tasks using the Web-based System Manager NIM interface (available in AIX 4.3 and later) or the System Management Interface Tool (SMIT), as well as from the command line. The procedures in this section identify prerequisite tasks or conditions that must be met before performing the task.

To start the Web-based System Manager NIM interface from an X-Windows session on the NIM master, enter:

```
wsm
```

Notes:

1. For additional information about NIM operations and the required and optional attributes to customize operations, see "Chapter 8. Network Installation Management Concepts" on page 73.
2. For assistance, use the online contextual help available for both the Web-based System Manager and SMIT interfaces.

   Extended help is available in the Web-based System Manager interface, offering guidance for tasks you may need to do in NIM. To view extended help, select **Contents** from the Help menu in the NIM container.
3. For tasks performed at the command line, the root user must be using **ksh**. Unexpected results can occur if the root user is set to another shell, such as **csh**.
4. If errors occur, it may be necessary to reset the machine before retrying the operation. For the procedure to reset a NIM client, see "Resetting Machines" on page 36.

Configuring the NIM Master and Creating Basic Installation Resources

Using this procedure, you can configure the NIM master, create the minimum basic installation resources required to install NIM client machines, and manage the resources for diskless and dataless clients.

Note: This procedure produces a large amount of output, especially when creating the **SPOT** resource. Be sure to scan through the output to look for nonfatal errors and warnings that may not be evident from a successful return code.

Prerequisites

The NIM master must have at least 750MB of available disk space. If such space is not available, see “Using Client Machines as Resource Servers” on page 37, “Defining /usr versus non-/usr SPOTs” on page 53, and “Defining an lpp_source on CD-ROM versus Disk” on page 55.

From Web-based System Manager

1. Insert the AIX 5.1 Server Product media into the appropriate drive of the designated master machine.
2. Start the Web-based System Manager Software application by entering `wsm`.
3. In the navigation area, expand and select the NIM container, then follow the directions in the wizard.

From SMIT

1. Insert the AIX 5.1 Server Product media into the appropriate drive of the designated master machine.
2. To install the **bos.sysmgt.nim** fileset, enter the **smit install_latest** fast path.
3. Using the LIST option, select **/dev/cd0** or **/dev/rmt0** for the INPUT device / directory for software.
4. Specify **bos.sysmgt.nim** as the SOFTWARE to install.
5. Accept the default values for all other fields on this screen. After successful completion of this installation, exit SMIT.
6. To configure the NIM master, enter the **smit nim_config_env** fast path.
7. Using the LIST option, select the Primary Network Interface for the NIM Master.
8. Using the LIST option, select **/dev/cd0** or **/dev/rmt0** for the `Input device for installation \ images` field.
9. If you will be supporting diskless and dataless clients, select **yes** at the `Create Diskless/Dataless Machine Resources?` field, and supply names for the resources to be created.
10. Select **yes** at the `Remove all newly added NIM definitions and file systems if any part of this operation fails?` field. This will make it easier to restart this procedure if failures occur.
11. Accept the default values for all other fields on this screen.

Notes:

- Depending on the speed of your machine, this could be a very lengthy process.
- This procedure provides the capability for much more than just configuring the NIM master and creating the **lpp_source** and **SPOT** resources. However, for this simple configuration, only a subset of the available functions will be used. Advanced NIM administrators can use the SMIT screens accessed through this procedure to create a more complex environment.
- As you develop a better understanding of configuration tasks, you may prefer to not automatically undo all configuration when failures occur (as in step 10 above). Continuing from the last point of failure results in faster configuration for experienced administrators.

From the Command Line

1. Insert the AIX 5.1 Server Product CD or tape into the CD-ROM or tape drive of the designated master machine.
2. If installing from a tape, skip to step 5. To create a mount point for the CD, enter:
   ```
   mkdir /cdfs
   ```
3. To create a **cdrom** file system, enter:
   ```
   crfs -v cdrfs -p ro -d'cd0' -m'/cdfs'
   ```
4. To mount the CD, enter:
   ```
   mount /cdfs
   ```
5. To install the **bos.sysmgt.nim** fileset from the CD, enter:
   ```
   installp -agX -d /cdfs/usr/sys/inst.images bos.sysmgt.nim
   ```

To install the **bos.sysmgt.nim** fileset from a tape, enter:

```
installp -agX -d /dev/rmt0 bos.sysmgt.nim
```

6. If installing from CD, to unmount the **cdrom** file system, enter:

   ```
   unmount /cdfs
   ```

7. To configure the NIM master using the **nimconfig** command, enter:

   ```
   nimconfig -a attr1=value1 \
   -a attr2=value2 \
   ...
   ```

   For example, to configure a NIM master with the following configuration:

   ```
   master host name = master1
   primary network interface = tr0
   ring speed = 16
   platform = rspc
   kernel type = mp
   ```

   Enter the following command sequence:

   ```
   nimconfig -a netname=network1 -a pif_name=tr0 \
   -a ring_speed1=16 -a platform=rspc -a netboot_kernel=mp
   ```

   Note:S ee the **nimconfig** command for additional attribute information.

8. To create a file system in the `rootvg` volume group with `400`MB of space with a mount point of `/export/lpp_source`, enter:

   ```
   crfs -v jfs -g rootvg -a size=$((2000*400)) \
   -m /export/lpp_source -A yes -p rw -t no \
   -a frag=4096 -a nbpi=4096 -a compress=no
   ```

9. To mount the file system, enter:

   ```
   mount /export/lpp_source
   ```

10. The **lpp_source** contains the installation images copied from the source device (in this example, the CD-ROM). The server of the **lpp_source** will be the NIM master. The images will be stored in the `/export/lpp_source/lpp_source1` directory. To create the `lpp_source` resource named `lpp_source1`, enter:

    ```
    nim -o define -t lpp_source -a source=/dev/cd0 \
    -a server=master -a location=/export/lpp_source/lpp_source1 \
    lpp_source1
    ```

11. To create a file system in the `rootvg` volume group with `200`MB of space with a mount point of `/export/spot`, enter:

    ```
    crfs -v jfs -g rootvg -a size=$((2000*200)) \
    -m /export/spot -A yes -p rw -t no \
    -a frag=4096 -a nbpi=4096 -a compress=no
    ```

12. To mount the file system, enter:

    ```
    mount /export/spot
    ```

13. The **SPOT** resource will be installed from images in the image source (in this case, the **lpp_source** that was created in step 10). The server of the resource will be the NIM master, and the **SPOT** will be stored in the `/export/spot/spot1` directory. To create the **SPOT** resource named `spot1`, enter:

    ```
    nim -o define -t spot -a source=lpp_source1 \
    -a server=master -a location=/export/spot spot1
    ```

14. If you are not supporting diskless and dataless clients, you do not need to continue with this procedure. If you are supporting diskless and dataless clients, create and mount a file system for their resources.

    To create a file system in the `rootvg` volume group with `150`MB of space and a mount point of `/export/dd_resource`, enter:

```
crfs -v jfs -g rootvg -a size=$((2000*150)) \
-m /export/dd_resource -A yes -p rw -t no \
-a frag=4096 -a nbpi=4096 -a compress=no
```

15. To mount the file system, enter:

```
mount /export/dd_resource
```

16. Create the diskless and dataless client resources in subdirectories of the **/export/dd_resource** directory. Not all resources are required. Create only the resources to be used in your environment.

    To create the `root` resource named `root1` (required for diskless and dataless clients), enter:

```
nim -o define -t root -a server=master \
-a location=/export/dd_resource/root1 root1
```

    To create the `dump` resource named `dump1` (required for diskless and dataless clients), enter:

```
nim -o define -t dump -a server=master \
-a location=/export/dd_resource/dump1 dump1
```

    To create the `paging` resource named `paging1` (required for diskless clients), enter:

```
nim -o define -t paging -a server=master \
-a location=/export/dd_resource/paging1 paging1
```

    To create the `home` resource named `home1` (optional), enter:

```
nim -o define -t home -a server=master \
-a location=/export/dd_resource/home1 home1
```

    To create the `shared_home` resource named `shared_home1` (optional), enter:

```
nim -o define -t shared-home -a server=master \
-a location=/export/dd_resource/shared_home1 shared_home1
```

    To create the `tmp` resource named `tmp1` (optional), enter:

```
nim -o define -t tmp -a server=master \
-a location=/export/dd_resource/tmp1 tmp1
```

    Notes:

    - The file systems created for the NIM resources are not required, but they can be beneficial for storage management.
    - A **SPOT** resource is required for supporting diskless and dataless clients. The **SPOT** created in step 13 can be used for this purpose, so there is no need to create another **SPOT** specifically for diskless and dataless clients.
    - For more information about NIM resources, see “NIM Resources” on page 83.

Adding a Standalone NIM Client to the NIM Environment

This procedure describes how to add standalone clients to the NIM environment. Standalone clients are machines that, once installed, can obtain a boot image and mount all file systems from the local hard disk, unlike diskless and dataless clients which depend on remote servers.

From one of the following interfaces, use Method A if the client machine is not running or if the client does not have AIX Version 4 installed. Method A can also be used if BOS is to be installed on the client and the client is to be network-booted manually or to initiate the install from a **force-push** operation. Prior to AIX 4.2, specific network objects must be defined in addition to the steps provided in Method A. This procedure automatically adds NIM networks when needed.

From one of the following interfaces, use Method B if the client machine has AIX AIX Version 4 already installed.

If the NIM client being defined is on a network that is not currently defined in the NIM environment, the **niminit** command will fail. If this is the case, use "Method A:" of this procedure to define the client on the NIM master, and then follow the steps in "Method B:" to complete the configuration.

Prerequisites

1. The NIM master must be configured. See "Configuring the NIM Master and Creating Basic Installation Resources" on page 7 for more information.
2. You must know the subnet mask, the default gateway for the client machine, and the default gateway for the NIM master.

From Web-based System Manager

1. To start the Web-based System Manager Software application, enter

   ```
   wsm
   ```
2. In the navigation area, expand and select the NIM container.
3. Select the Machines container.
4. From the Machines menu, select **New** —> **Machine**.
5. Use the wizard to complete the task.

From SMIT

Method A:

1. To add a standalone client to the NIM environment, enter the **smit nim_mkmac** fast path.
2. Specify the host name of the client.
3. The next SMIT screen displayed depends on whether NIM already has information about the client's network. Supply the values for the required fields or accept the defaults. Use the help information and the LIST option to help you specify the correct values to add the client machine.

Method B:

1. Install the **bos.sysmgt.nim.client** fileset on the running machine.
2. Enter the **smit niminit** fast path.
3. Supply the values for the required fields or accept the defaults. Use the help information and the LIST option to help you specify the correct values for defining your client machine.

   Note: If the LIST option is used to display valid platforms for the client definition, only platforms currently supported by **SPOTs** defined in the NIM environment are displayed. If no **SPOTs** are currently defined, only **rs6k** and **rspc** are displayed as selectable platforms.

From the Command Line

Method A:

On the NIM master, enter:

```
nim -o define -t standalone -a
platform=PlatformType \
-a netboot_kernel=NetbootKernelType \
-a if1=InterfaceDescription \
-a net_definition=DefinitionName -a ring_speed1=SpeedValue \
-a cable_type1=TypeValue -a iplrom_emu=DeviceName MachineName
```

Example 1:

To add the machine with host name `machine1` with the following configuration:

```
host name=machine1
platform=rspc
kernel=up
network type=ethernet
subnet mask=255.255.240.0
default gateway=gw1
default gateway used by NIM master=gw_master
cable type=bnc
network boot capability=yes (no emulation needed)
```

enter the following command sequence:

```
nim -o define -t standalone -a
platform="rspc" \
-a netboot_kernel="up" -a if1="find_net machine1 0" \
-a cable_type1="bnc" \
-a net_definition="ent 255.255.240.0 gw1 gw_master" machine1
```

Example 2:

To add the machine with host name `machine2` with the following configuration:

```
host name=machine2
platform=rs6k
netboot_kernel=up
network type=token ring
subnet mask=255.255.225.0
default gateway=gw2
default gateway used by NIM master=gw_master
ring speed=16
network boot capability=no (use emulation on a diskette)
```

enter the following command sequence:

```
nim -o define -t standalone -a
platform="rs6k" \
-a netboot_kernel="up" -a if1="find_net machine2 0" \
-a ring_speed1="16" \
-a net_definition="tok 255.255.225.0 gw2 gw_master" \
-a iplrom_emu="/dev/fd0" machine2
```

Notes:

1. Prior to AIX 4.2, specific network objects must be defined in addition to the steps provided in this procedure. In this procedure, NIM networks are added automatically when needed.
2. If the **find_net** keyword in the **if** attribute causes NIM to successfully match a network definition to the client definition, the **net_definition** attribute is ignored.
3. See “Defining NIM Clients” on page 74 for more information about the attributes you can specify when defining NIM clients.

Method B:

1. Install the **bos.sysmgt.nim.client** fileset on the client machine.
2. From the machine being defined as a client, enter:

```
niminit -a name=ClientDefinitionName -a
master=MasterName \
-a pif_name=Interface -a platform=PlatformType \
-a netboot_kernel=NetbootKernelType -a ring_speed1=SpeedValue \
-a cable_type1=TypeValue -a iplrom_emu=DeviceName
```

Note: For detailed attribute information, see the **niminit** command.

Example 1:

To add the machine with host name machine1 with the following configuration:

```
host name=machine1
NIM master's host name=master_mac
primary interface adapter=en0
platform=rspc
kernel=up
cable type=bnc
network boot capability=yes (no emulation needed)
```

enter the following command sequence:

```
niminit -a name=machine1 -a
master=master_mac \
-a pif_name=en0 -a platform=rspc -a netboot_kernel=up \
-a cable_type1=bnc
```

Example 2:

To add the machine with host name machine2 with the following configuration:

```
host name=machine2
NIM master's host name=master_mac
primary interface adapter=tr0
platform=rs6k
netboot_kernel=up
ring speed1=16
network boot capability=no (use emulation on a diskette)
```

enter the following command sequence:

```
niminit -a name=machine2 -a
master=master_mac \
-a pif_name=tr0 -a platform=rs6k -a netboot_kernel=up \
-a ring_speed1=16 -a iplrom_emu="/dev/fd0"
```

Verifying the Status of Your Client Machine

To verify that the **niminit** operation completed successfully, enter the following command at the NIM client:

```
nimclient -l -l MachineObjectName
```

Note: There is neither a Web-based System Manager application nor a SMIT menu to do this task.

The system returns output similar to the following:

```
Standalone2:
   class          =  machines
   type           =  standalone
   Cstate         =  ready for a NIM operation
   platform       =  rs6k
   netboot_kernel =  up
   if1            =  Network2 standalone2 08005acd536d
   cable_type1    =  bnc
   iplrom_emu     =  /dev/fd0
   prev_state     =  customization is being performed
   cpuid          =  000247903100
   Mstate         =  currently running
   Cstate_result  =  success
```

If the system output to this query indicates any errors, you must validate all of your data, checking for accurate spelling, nonduplication of NIM names, and so forth, and redo the **niminit** operation.

Be sure to coordinate this operation with the system administrator of the NIM master, and ensure that *all* NIM object names are unique in the entire NIM environment.

Using Installation Images to Install the Base Operating System (BOS) on a NIM Client (rte Install)

Using installation images to install BOS on a NIM client is similar to the traditional BOS installation from a tape or CD-ROM device, because the BOS image is installed from the installation images in the **lpp_source** resource.

Prerequisites

- The NIM master must be configured, and **lpp_source** and **SPOT** resources must be defined. See "Configuring the NIM Master and Creating Basic Installation Resources" on page 7.
- The NIM client to be installed must already exist in the NIM environment. To add the client to the NIM environment, use the "Adding a Standalone NIM Client to the NIM Environment" on page 10 procedure.

From Web-based System Manager

1. From the NIM container, select the Machines container.
2. In the contents area, select a target standalone machine for the installation.
3. From the Selected menu, choose **Install Operating System**.
4. If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. If the client was not rebooted automatically, initiate a network boot from the client to install it. Follow the "Booting a Machine Over the Network" on page 159 procedure to perform the network boot.
5. After the machine boots over the network, the display on the client machine will begin prompting for information about how to configure the machine during installation. Specify the requested information to continue with the installation.

   Note: To perform a non-prompted installation, follow the instructions in "Performing a Non-Prompted BOS Installation" on page 20 to complete the prerequisite tasks.

From SMIT

1. To install BOS on a NIM client using an **rte** install, enter **smit nim_bosinst** from the NIM master.
2. Select the TARGET for the operation.
3. Select **rte** as the installation TYPE.
4. Select the SPOT to use for the installation.
5. Select the LPP_SOURCE to use for the installation.
6. In the displayed dialog fields, supply the correct values for the installation options or accept the default values. Use the help information and the LIST option to help you.
7. If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. If the client was not rebooted automatically from SMIT, initiate a network boot from the client to install it. Use the procedure for "Booting a Machine Over the Network" on page 159 to initiate the network boot.
8. After the machine boots over the network, the display on the client machine will begin prompting for information about how the machine should be configured during installation. Specify the requested information to continue with the installation.

   Note: To perform a non-prompted installation, follow the instructions in "Performing a Non-Prompted BOS Installation" on page 20 to complete the prerequisite tasks.

From the Command Line

1. To initiate the **bos_inst** operation, enter:

```
nim -o bos_inst -a source=rte -a lpp_source=Lpp_Source \
-a spot=SPOTName -a accept_licenses=yes -a boot_client=yes/no ClientName
```

Specify the resources to be used to support the installation and any additional options for customizing the installation. To perform a simple **rte** installation, specify the **lpp_source** and **SPOT** resources.

If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. A network boot must be performed manually on the machine. If that is the case, supply the **boot_client=no** attribute to the **bos_inst** command. If the **boot_client** attribute value is not specified, it defaults to **boot_client=yes**.

2. If the client was not rebooted automatically, initiate a network boot from the client to install it. Follow the “Booting a Machine Over the Network” on page 159 procedure to perform the network boot.
3. After the machine boots over the network, the display on the client machine will begin prompting for information about how to configure the machine during installation. Specify the requested information to continue with the installation.

   Example

   The client machine, `machine1`, is not a running, configured NIM client. You should specify **boot_client=no**. To install the client using the **lpp_source** named `lpp_source1` and the **SPOT** named `spot1`, enter:

   ```
   nim -o bos_inst -a source=rte -a lpp_source=lpp_source1 \
   -a spot=spot1 -a accept_licenses=yes -a boot_client=no machine1
   ```

   Notes:

   a. The steps to perform an **rte** installation are almost identical to the steps to perform other types of BOS installations. The main difference is that **rte** must be specified in the **source** attribute of the **nim bos_inst** command.
   b. To perform a non-prompted installation, follow the instructions in “Performing a Non-Prompted BOS Installation” on page 20 to complete the prerequisite tasks.
   c. For a complete description of the different ways that a BOS installation can be customized by NIM, see the “bos_inst” on page 104 operation.

Using a mksysb Image to Install the Base Operating System (BOS) on a NIM Client (mksysb Install)

A **mksysb** installation restores BOS and additional software to a target from a **mksysb** image in the NIM environment.

For a complete description of different ways to customize a BOS installation using NIM, see the “bos_inst” on page 104 operation.

Prerequisites

1. The NIM master must be configured, and **lpp_source** and **SPOT** resources must be defined. See “Configuring the NIM Master and Creating Basic Installation Resources” on page 7.
2. The NIM client to be installed must already exist in the NIM environment. To add the client to the NIM environment, use the “Adding a Standalone NIM Client to the NIM Environment” on page 10 procedure.
3. The **mksysb** must be available on the hard disk of the NIM master or a running NIM client, or the **mksysb** image will be created during this procedure from either the NIM master or a running NIM client.
4. Although existing AIX 4.1.5 **mksysb** images can be used to install BOS, only NIM clients running AIX 4.2 for later can be used as source machines when creating **mksysb** images for resources.
5. The **SPOT** and **mksysb** resources should be at the same level of AIX when used for NIM BOS installations.

6. Many applications, particularly databases, maintain data in *sparse files*. A sparse file is one with empty space, or gaps, left open for future addition of data. If the empty spaces are filled with the ASCII null character and the spaces are large enough, the file will be sparse, and disk blocks will not be allocated to it.

   This creates an exposure in that a large file will be created, but the disk blocks will not be allocated. As data is then added to the file, the disk blocks will be allocated, but there may not be enough free disk blocks in the file system. The file system can become full, and writes to any file in the file system will fail.

   It is recommended that you either have no sparse files on your system or that you ensure you have enough free space in the file system for future allocation of the blocks.

Cloning Considerations

The **mksysb** images enable you to clone one system image onto multiple target systems. The target systems might not contain the same hardware devices or adapters, require the same kernel (uniprocessor or multiprocessor), or be the same hardware platform (rs6k, rspc, or chrp) as the source system.

Because NIM configures TCPIP at the end of an install, it is recommended that a **bosinst_data** resource be allocated for cloning **mksysb** installs with the field RECOVER_DEVICES set to `no`. This will prevent the BOS install process from attempting to configure the devices as they were on the source machine of the **mksysb**.

> **Attention:** If the system you have cloned is using OpenGL or graPHIGS, there may be some device filesets from these LPPs that must be installed after a clone. OpenGL and graPHIGS have graphics adapter-specific filesets, so if you cloned onto a system with a different graphics adapter, you will need to create a bundle as follows:
>
> ```
> echo OpenGL.OpenGL_X.dev >
>   /usr/sys/inst.data/user_bundles/graphic_dev.bnd
> echo PEX_PHIGS.dev >>
>   /usr/sys/inst.data/user_bundles/graphic_dev.bnd
> ```
>
> You can allocate this bundle when you install the **mksysb**, and the device filesets will be installed automatically if OpenGL and graPHIGS are in your **lpp_source**.

From Web-based System Manager

1. In the NIM container, select the Resources container.
2. From the Resources menu, select **New —> Resource —> New Resources**.
3. Use the wizard to complete the task.

> **Note:** To perform a non-prompted installation, follow the instructions in “Performing a Non-Prompted BOS Installation” on page 20 to complete the prerequisite tasks.

From SMIT

1. If the **mksysb** resource has already been created, skip to step 6. Otherwise, to create the **mksysb** resource, enter the **smit nim_mkres** fast path.
2. Select **mksysb** from the list of resource types that can be defined.
3. In the displayed dialogs, supply the values for the required fields. Use the help information and the **LIST** option to help you specify the correct values for defining your **mksysb** resource.
4. If the **mksysb** image does not exist, create it by supplying the values for the fields under **System Backup Image Creation Options**.

   > **Note:** If the **mksysb** image already exists as a file on the hard disk of the NIM master or client, no additional information is needed to define your **mksysb** resource.
5. Upon successful completion of this task, exit SMIT.

6. To use the **mksysb** resource to install a NIM client, enter the **smit nim_bosinst** fast path.
7. Select a TARGET for the operation.
8. Select **mksysb** as the installation TYPE.
9. Select the MKSYSB to use for the installation.
10. Select the SPOT to use for the installation.
11. Select the LPP_SOURCE to use for the installation.
12. In the displayed dialog fields, supply the correct values for the installation options or accept the default values. Use the help information or the LIST option to help you.
13. Run the SMIT dialog to install the NIM client.
14. If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. If the client was not rebooted automatically from SMIT, initiate a network boot from the client to install it. Use the "Booting a Machine Over the Network" on page 159 procedure to initiate the network boot.
15. After the machine boots over the network, the display on the client machine will begin prompting for information about how to configure the machine during installation. Specify the requested information to continue with the installation.

    Note: To perform a non-prompted installation, follow the instructions in "Performing a Non-Prompted BOS Installation" on page 20 to complete the prerequisite tasks.

From the Command Line

1. If the **mksysb** resource has already been created, skip to step 2. To create the **mksysb** resource, enter:

   ```
   nim -o define -t mksysb -a server=ServerName \
   -a location=LocationName -a mk_image=yes \
   -a source=SourceMachine ResourceName
   ```

   Specify the server name and location of the **mksysb** image. The **mk_image** and **source** attributes are used to create the **mksysb** image if it does not already exist.

   For a complete description of all the options that can be specified when creating a **mksysb** resource, see "mksysb Resource" on page 91.

   Example 1:

   To define a **mksysb** resource, `mksysb_res1`, from an existing **mksysb** image located in `/export/backups/client_mksysb` on the master, enter:

   ```
   nim -o define -t mksysb -a server=master \
   -a location=/export/backups/client_mksysb mksysb_res1
   ```

   Example 2:

   To create a **mksysb** image of the client machine, `client1`, in `/export/resources/new_mksysb` on the master, and to define a **mksysb** resource, `mksysb_res2`, enter:

   ```
   nim -o define -t mksysb -a server=master \
   -a location=export/resources/new_mksysb -a mk_image=yes \
   -a source=client1 mksysb_res2
   ```

2. To initiate the **bos_inst** operation, enter:

   ```
   nim -o bos_inst -a source=mksysb -a mksysb=mksysb \
   -a lpp_source=Lpp_Source -a spot=SPOTName \
   -a boot_client=yes/no ClientName
   ```

Specify the resources to be used to support the installation and any additional options for customizing the installation. To perform a simple **mksysb** installation, specify the **mksysb**, **lpp_source**, and **SPOT** resources.

If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. A network boot must be performed manually on the machine. If that is the case, supply the **boot_client=no** attribute to the **bos_inst** command. If the **boot_client** attribute value is not specified, it defaults to **boot_client=yes**.

3. If the client was not rebooted automatically, initiate a network boot from the client to install it. Follow the “Booting a Machine Over the Network” on page 159 procedure to perform the network boot.
4. After the machine boots over the network, the display on the client machine will begin prompting for information about how to configure the machine during installation. Specify the requested information to continue with the installation.

   Example 3:

   To perform a **mksysb** installation using the **mksysb**, `mksysb1`, the **lpp_source**, `lpp_source1`, and the **SPOT**, `spot1`, on client machine, `machine1`, which is not a running, configured NIM client, enter:

   ```
   nim -o bos_inst -a source=mksysb -a mksysb=mksysb1 \
   -a lpp_source=lpp_source1 -a spot=spot1 -a boot_client=no machine1
   ```

   Notes:

   1. The steps to perform a **mksysb** installation are almost identical to the steps to perform other types of BOS installations. The main differences are that **mksysb** must be specified in the **source** attribute of the **nim bos_inst** command, and a **mksysb** resource must be allocated for the operation.
   2. To perform a non-prompted installation, follow the instructions in “Performing a Non-Prompted BOS Installation” on page 20 to complete the prerequisite tasks.

Using a SPOT-copy to Install the Base Operating System (BOS) on a NIM Client (SPOT-copy Install)

A **SPOT-copy** installs the BOS image on a machine by copying the BOS files from a **SPOT** resource.

For a complete description of different ways to customize a BOS installation using NIM, see “bos_inst” on page 104 operation .

Prerequisites

1. The NIM master must be configured, and **lpp_source** and **SPOT** resources must be defined. See “Configuring the NIM Master and Creating Basic Installation Resources” on page 7.
2. The NIM client to be installed must already exist in the NIM environment. To add the client to the NIM environment, use the “Adding a Standalone NIM Client to the NIM Environment” on page 10 procedure.

From Web-based System Manager

1. From the NIM Container, select the Machines container.
2. In the contents area, select a target standalone machine for the install.
3. From the Selected menu, choose **Install Operating System**.
4. In the dialog, under Installation type, select **spot - Perform a SPOT Copy Using a SPOT image** and fill in the required fields.
5. If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. If the client was not rebooted automatically, initiate a network boot from the client to install it. Follow the “Booting a Machine Over the Network” on page 159 procedure to perform the network boot.

6. After the machine boots over the network, the display on the client machine will begin prompting for information about how to configure the machine during installation. Specify the requested information to continue with the installation.

   Note: To perform a non-prompted installation, follow the instructions in “Performing a Non-Prompted BOS Installation” on page 20 to complete the prerequisite tasks.

From SMIT

1. To install BOS on a NIM client using a **SPOT-copy** install, enter **smit nim_bosinst** from the NIM master.
2. Select the TARGET for the operation.
3. Select **spot** as the installation TYPE.
4. Select the SPOT to use for the installation.
5. Select the LPP_SOURCE to use for the installation.
6. In the displayed dialog fields, supply the correct values for the installation options or accept the default values. Use the help information and the LIST option to help you.
7. Run the SMIT dialog.
8. If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. If the client was not rebooted automatically, initiate a network boot from the client to install it. Follow the “Booting a Machine Over the Network” on page 159 procedure to perform the network boot.
9. After the machine boots over the network, the display on the client machine will begin prompting for information about how to configure the machine during installation. Specify the requested information to continue with the installation.

   Note: To perform a non-prompted installation, follow the instructions in “Performing a Non-Prompted BOS Installation” on page 20 to complete the prerequisite tasks.

From the Command Line

1. To initiate the **bos_inst** operation, enter:

   ```
   nim -o bos_inst -a source=spot -a
   lpp_source=Lpp_Source \
   -a spot=SPOTName -a accept_licenses=yes -a boot_client=yes/no ClientName
   ```

   Specify the resources to be used to support the installation and any additional options for customizing the installation. To perform a simple **SPOT-copy** installation, specify the **lpp_source** and **SPOT** resources.

   If the client machine being installed is not already a running, configured NIM client, NIM will not automatically reboot the machine over the network for installation. A network boot must be performed manually on the machine. If that is the case, supply the **boot_client=no** attribute to the **bos_inst** command. (If the **boot_client** attribute value is not specified, it defaults to **boot_client=yes**.)
2. If the client was not rebooted automatically, initiate a network boot from the client to install it. Follow the “Booting a Machine Over the Network” on page 159 procedure to perform the network boot.
3. After the machine boots over the network, the display on the client machine will begin prompting for information about how to configure the machine during installation. Specify the requested information to continue with the installation.

   Example:

   The client machine, `machine1`, is not a running, configured NIM client. You should specify **boot_client=no**. To install the client using the **lpp_source** named `lpp_source1` and the **SPOT** named `spot1`, enter:

```
nim -o bos_inst -a source=spot -a
lpp_source=lpp_source1 \
-a spot=spot1 -a accept_licenses=yes -a boot_client=no machine1
```

Notes:

1. The steps to perform a **SPOT-copy** installation are almost identical to the steps to perform other types of BOS installations. The main difference is that **spot** must be specified as the **source** attribute of the **nim bos_inst** command.
2. To perform a non-prompted installation, follow the instructions in "Performing a Non-Prompted BOS Installation" to complete the prerequisite tasks.

Performing a Non-Prompted BOS Installation

Prerequisites

1. The NIM master must be configured, and **lpp_source** and **SPOT** resources must be defined. See "Configuring the NIM Master and Creating Basic Installation Resources" on page 7.
2. The NIM client to be installed must already exist in the NIM environment. To add the client to the NIM environment, use the procedure "Adding a Standalone NIM Client to the NIM Environment" on page 10.
3. If any of the software to be installed during the BOS installation requires acceptance of a license agreement, determine whether to accept the license agreement during BOS installation or defer acceptance until after the client has booted. Note that license acceptance takes place at the client. See 165 for a sample bosinst.data file that specifies the syntax to control license acceptance.

From Web-based System Manager

You can use the Install Base OS wizard to create a **bosinst_data** resource. The **bosinst_data** resource may be used for a **rte** Install or a **mksysb** Install.

If you want to create the **bosinst_data** resource or do a **SPOT**-copy installation, continue with the following steps:

1. In the Resources container, from the Resources menu, select **New —> Resource**. The Add New Resource wizard displays.
2. Follow the wizard instructions to create a **bosinst_data** resource. The wizard creates a basic **bosinst.data** file, which can be used ″as is″ or can be enhanced according to sample files. See "Appendix C. Sample Files" on page 165 for a sample **bosinst.data** file. To do a nonprompted installation, the **bosinst_data** resource must be created first.
3. Continue with the installation by following one of the install procedures:
   - "Using Installation Images to Install the Base Operating System (BOS) on a NIM Client (rte Install)" on page 14
   - "Using a mksysb Image to Install the Base Operating System (BOS) on a NIM Client (mksysb Install)" on page 15
   - "Using a SPOT-copy to Install the Base Operating System (BOS) on a NIM Client (SPOT-copy Install)" on page 18

From SMIT

1. On the NIM master or any running NIM client, create a **bosinst.data** file that describes how a machine should be configured during a BOS installation. See "Appendix C. Sample Files" on page 165 for a sample **bosinst.data** file.
2. To define the **bosinst.data** file as a **bosinst_data** resource in the NIM environment, enter the **smit nim_mkres** fast path.
3. Select **bosinst_data** from the list of resource types displayed on your screen.

4. Supply the values for the required fields. Use the help information and the LIST option to help you specify the correct values for defining your **bosinst_data** resource.
5. After the **bosinst_data** resource has been defined, follow the procedures for performing an **rte**, **mksysb**, or **SPOT-copy** installation on a standalone machine. Be sure to specify the **bosinst_data** resource to use during the installation.

From the Command Line

1. On the NIM master or any running NIM client, create a **bosinst.data** file that describes how a machine should be configured during a BOS installation. See “Appendix C. Sample Files” on page 165 for a sample **bosinst.data** file.

   Note: To accept license agreements for software to be installed during the BOS installation, specify `-a accept_licenses=yes` on the `nim -o bos_inst` command.
2. To define the **bosinst.data** file as a **bosinst_data** resource, enter:

   ```
   nim -o define -t bosinst_data -a server=ServerName \
   -a location=LocationName NameValue
   ```

   Using the **server** attribute, specify the name of the machine where the **bosinst.data** file is located.

   Using the **location** attribute, specify the full path name of the **bosinst.data** file that is to be used as a resource.
3. After the **bosinst_data** resource has been defined, follow the normal procedure for performing an **rte**, **mksysb**, or **SPOT-copy** installation on standalone machines. Be sure to specify that the **bosinst_data** resource should be used for the installation.

   For example, to perform a non-prompted **rte** install of `machine1` using the `lpp_source1`, `spot1`, and `bosinst_data1` resources, enter:

   ```
   nim -o bos_inst -a source=rte -a lpp_source=lpp_source1 \
   -a spot=spot1 -a accept_licenses=yes -a bosinst_data=bosinst_data1 \
    machine1
   ```

Installing to Clients on ATM Networks

Unlike other network adapters, ATM adapters cannot be used to boot a machine. This means that installing a machine over an ATM network requires special processing.

BOS Installation Over Non-ATM Adapters

Normally when a machine performs a network boot over a specified adapter, the adapter is configured by IPL-ROM or firmware. Then a boot image is transferred from the boot server to the client using tftp. This boot image performs further configuration and mounts network installation resources before starting the BOS installation.

BOS Installation Over ATM Adapters

Since an ATM adapter cannot be configured by IPL-ROM or firmware, a boot image cannot be obtained over the network to perform a BOS installation. This means that the NIM bos_inst operation must copy a boot image to the hard disk of the client before the machine is rebooted. Some ODM information is also saved on the client machine so that when the machine is rebooted, the ATM adapter can be configured properly.

NIM clients may not have the executables installed to support the special processing required for installation over ATM, so the directories **/usr/lib/boot/bin** and **/usr/lpp/bos.sysmgt/nim/methods** are mounted at the client from the NIM master. These directories contain executables that are run during the setup performed by the NIM bos_inst operation.

After the initial setup completes, an ″at″ job is issued to reboot the machine after one minute has elapsed. When the machine reboots, the boot image that was copied to the hard disk configures the ATM adapter and mounts network installation resources for the BOS installation. The installation then proceeds as normal until the customization phase. During NIM customization, the ATM adapter is not reconfigured with a **mktcpip** command since the ODM already contains information carried over from before the machine was reinstalled. All other aspects of NIM customization are the same as for non-ATM clients.

Configuration Requirements

- Machines that will have BOS installed over ATM must be running and configured NIM clients.

  Note: Configured NIM clients have the **bos.sysmgt.nim.client** fileset installed, are registered in the NIM master database, and have a valid **/etc/niminfo** file.
- BOS installations over ATM adapters will always use the **at0** interface on the client.
- The NIM master fileset must be installed at AIX 4.3.0 with the update for ATM install or any superseding level.
- The SPOT that will be used to install the clients must be at version 4.3.0.0 with the update for ATM install or any superseding level.
- The NIM master must be installed with the base device filesets to support the platforms of clients that will be installed. The following table shows the filesets that must be installed on the NIM master to support different platforms:

rs6k	**devices.base.rte**
rspc	**devices.rspc.base.rte**
chrp	**devices.chrp.base.rte**

Converting Generic Networks Into ATM Networks

Prior to the support of BOS installations over ATM, it was necessary to define ATM networks as ″generic″ networks for performing other types of NIM operations. To convert generic networks into ATM networks, enter the following command:

nim -o change -a new_type=atm (network)

The adapter names for the client interfaces on the ATM network will automatically be set to **at0** in the NIM database.

If desired, the name of the network can be changed also:

nim -o change -a new_name=(new network name) (current network name)

System Recovery After Boot Failure

Since BOS installation over ATM requires a special boot image to be written to the hard disk of the client, the original boot image on the machine will be lost. This means that if the installation is aborted or fails before BOS is reinstalled, it will not be possible to perform a normal reboot of the client unless system maintenance is performed. By performing system maintenance, a new boot image can be created on the hard disk to allow the machine to be booted for normal use. The following procedure should be used:

1. Boot the client from a CD.
2. When the installation options are displayed, select the option to perform system maintenance.
3. Make the necessary selections to access the machine's root volume group.
4. In the maintenance shell, run the following sequence of commands:
   a. **bosboot -ad /dev/ipldevice**

b. **BLVDISK='lslv -l hd5 | grep hdisk | head -1 |cut -d' ' -f1'**

c. **bootlist -m normal $BLVDISK**

d. **sync**

e. **sync**

f. **sync**

g. **reboot -q**

If errors are detected during the NIM **bos_inst** operation and the client machines hasn't rebooted, it is possible to stop the machine from rebooting, and then execute the sequence of commands in the above step 4 on the running system. To stop the reboot, use the following procedure:

1. List the ″at″ jobs on the machine by entering the command: **at -1**

   The first name in the output field will be the name of the job.

2. To remove the ″at″ job, enter the following command: **at -r (name of job)**

   Example: **$ at -1 root.884205595.a Wed Jan 7 14:39:55 1998 $ at -r root.884205595.a at file: root.884205595.a deleted**

   Note: The reboot can also be prevented by removing the shutdown script that the ″at″ job was instructed to run by entering: **rm/tmp/_NIM_shutdown**

Customizing NIM Clients and SPOT Resources

This procedure describes how to use NIM to install software on running, configured NIM clients and **SPOT** resources.

Prerequisites

1. If the software is to be installed on a machine, the machine must be a running, configured NIM client with push permissions enabled for the NIM master. Push permissions are enabled by default when a client is configured or installed by NIM.
2. If the software is to be installed on a **SPOT** resource, the server of the **SPOT** must be running.
3. The installation image to be installed on the target is available in an **lpp_source** resource, and a **check** operation was performed on the **lpp_source** at some point after the image was first copied there. (The **check** operation updates the **.toc** file with information about the images present in the **lpp_source**.)

From Web-based System Manager

1. From the NIM container, select the Machines container.
2. In the contents area, select a target machine (master or standalone), or in the Resources container, select a target **SPOT**.
3. From the Selected menu, choose **Install/Update Software** —> **Install Additional Software (Custom)** to display the Install Software dialog.
4. Use the dialog to complete the task.

From SMIT

Several SMIT screens are available to make NIM installation operations easier to perform. The SMIT screens follow the same structure as those used for local installation operations performed on a system. When performing NIM customization operations, select the SMIT screen that most closely describes the installation you want to perform.

1. From the command line, enter the **smit nim_task_inst** fast path.
2. Select the SMIT menu item that matches the type of installation you want to perform.

3. Select a TARGET for the operation.
4. Select the **lpp_source** that contains the installation images to be used.
5. Select any other required resources.
6. In the final SMIT dialog, supply the values for the required fields or accept the defaults. Use the help information and the LIST option to help you specify the correct values.

From the Command Line

To perform the installation operation, enter:

```
nim -o cust -a lpp_source=Lpp_Source -a
filesets=FilesetsList \
-a installp_bundle=InstallpBundle \
-a installp_flags=InstallpFlags TargetName
```

You will specify the resources to use to support the installation and any additional attributes for customization.

The software to be installed on the client can be specified on the command line using either the **filesets** attribute or by specifying an **installp_bundle** resource that lists the software.

The default **installp** flags to be used to install the software are **-a**, **-g**, **-Q**, and **-X**. To specify a different set of **installp** flags, you can list them in the **installp_flags** attribute.

Example 1:

To install the **bos.diag** and **bos.dosutil** filesets on the client, `machine1`, using the **lpp_source** resource named `lpp_source1`, enter:

```
nim -o cust -a lpp_source=lpp_source1 \
-a filesets="bos.diag bos.dosutil" machine1
```

Example 2:

To install software into the **SPOT** resource, `spot1`, using the **lpp_source** resource, `lpp_source1`, and the list of filesets specified in the **installp_bundle** resource, `installp_bundle1`, enter:

```
nim -o cust -a lpp_source=lpp_source1 \
-a installp_bundle=installp_bundle1 spot1
```

> **Note:** Several other resources and attributes can be specified on the command line with the **cust** operation. See “NIM Operations” on page 101 for a complete description of the **cust** operation.

Configuring the NIM Master and Creating Resources to Support Diskless and Dataless Clients Only

Use this procedure only if the NIM environment is to be used exclusively for diskless and dataless client management. If the NIM environment is also to be used for installing and maintaining software on standalone machines, follow the procedure for “Configuring the NIM Master and Creating Basic Installation Resources” on page 7.

> **Note:** This procedure produces a large amount of output, especially when creating the **SPOT** resource. Be sure to scan through the output to look for nonfatal errors and warnings that may not be evident from a successful return code.

Prerequisites

The NIM master must have at least 300MB of available disk space. If such space is not available, see “Using Client Machines as Resource Servers” on page 37, “Defining /usr versus non-/usr SPOTs” on page 53, and “Defining an lpp_source on CD-ROM versus Disk” on page 55.

From Web-based System Manager

1. Insert the AIX 4.3 media into the appropriate drive of the designated master machine.
2. Start the Web-based System Manager application by entering **wsm**.
3. In the navigation area, select and expand the Software container.
4. While still in the navigation area, select the Installed Software container.
5. From the Software menu, choose **New Software (Install/Update)** —> **Install Additional Software**.
6. In the Install Software dialog, select `/dev/cd0` or `/dev/rmt0` as the software source.
7. Specify `bos.sysmgt.nim` as the software to install.
8. In the navigation area, select the NIM container.
9. From the NIM menu, select **Configure Environment**.
10. Follow the wizard instructions to guide you through the configuration.

From SMIT

1. Insert the AIX 4.2 (or later) Server Product CD or tape into the CD-ROM or tape drive of the designated master machine.
2. To install the **bos.sysmgt.nim** fileset, enter the **smit install_latest** fast path.
3. Using the LIST option, select **/dev/cd0** or **/dev/rmt0** for the INPUT device / directory for software.
4. Specify **bos.sysmgt.nim** as the SOFTWARE to install.
5. Accept the default values for all other fields on this screen. After completion of this installation, exit SMIT.
6. To configure the NIM master, enter the **smit nimconfig** fast path.
7. Specify a name in the Network Name field to be assigned to the NIM master’s network.
8. Using the LIST option, select the Primary Network Interface for the NIM Master.
9. Accept the default values for all other fields on this screen.
10. After the master is configured, exit SMIT.
11. Restart SMIT using the **smit nim_mkres_dd_name_server** fast path.
12. When prompted, select the NIM master as the server of the client resources.
13. Select **yes** in the `Create a new SPOT?` field, since there is not a **SPOT** currently defined in your environment.
14. Using the LIST option, select **/dev/cd0** or **/dev/rmt0** as the input device for installation images.
15. Specify a name in the SPOT Name field.
16. Specify names for the other resources to be created in the NIM environment. If a name is not specified, the resource will not be created.
17. Select **yes** at the `Remove all newly added NIM definitions and file systems if any part of this operation fails?` field. This will make it easier to restart this procedure if failures occur.
18. Accept the default values for all other fields on this screen.

> **Note:** In most NIM environments, the **SPOT** will already exist to support base operating system installation operations on standalone machines. In such environments, it is not necessary to create a new **SPOT**.

From the Command Line

1. Insert the AIX 4.2 (or later) Server Product CD or tape into the CD-ROM or tape drive of the designated master machine.
2. If installing from a tape, skip to step 5. To create a mount point for the CD, enter:

   ```
   mkdir /cdfs
   ```
3. To create a **cdrom** file system, enter:

   ```
   crfs -v cdrfs -p ro -d'cd0' -m'/cdfs'
   ```
4. To mount the CD, enter:

   ```
   mount /cdfs
   ```
5. To install the **bos.sysmgt.nim** fileset from the CD, enter:

   ```
   installp -agX -d /cdfs/usr/sys/inst.images
   bos.sysmgt.nim
   ```

   or to install the **bos.sysmgt.nim** fileset from a tape, enter:

   ```
   installp -agX -d /dev/rmt0
   bos.sysmgt.nim
   ```
6. If installing from CD, to unmount the **cdrom** file system, enter:

   ```
   unmount /cdfs
   ```
7. To configure the NIM master using the **nimconfig** command, enter:

   ```
   nimconfig -a attr1=value1 \
             -a attr2=value2 \
             ...
   ```

   For example, to configure a NIM master with the following configuration:

   ```
   master host name = master1
   primary network interface = tr0
   ring speed = 16
   platform = rspc
   kernel type = mp
   ```

   enter the following command sequence:

   ```
   nimconfig -a netname=network1 -a pif_name=tr0
   -a ring_speed=16 \
   -a platform=rspc -a netboot_kernel=mp
   ```

   Note: See the **nimconfig** command for additional attribute information.
8. To create a file system in the `rootvg` volume group with `200`MB of space and a mount point of `/export/spot`, enter:

   ```
   crfs -v jfs -g rootvg -a size=$((2000*200)) \
   -m /export/spot -A yes -p rw -t no \
   -a frag=4096 -a nbpi=4096 -a compress=no
   ```
9. To mount the file system, enter:

   ```
   mount /export/spot
   ```
10. The **SPOT** resource will be installed from images in the image source (in this example, the CD). The server of the resource will be the NIM master, and the **SPOT** will be stored in the `/export/spot/spot1` directory. To create the **SPOT** resource, enter:

    ```
    nim -o define -t spot -a source=/dev/cd0 -a
    server=master \
    -a location=/export/spot spot1
    ```
11. To create a file system in the `rootvg` volume group with `150`MB of space and a mount point of `/export/dd_resource`, enter:

```
crfs -v jfs -g rootvg -a size=$((2000*150)) \
-m /export/dd_resource -A yes -p rw -t no \
-a frag=4096 -a nbpi=4096 -a compress=no
```

12. To mount the file system, enter:

```
mount /export/dd_resource
```

13. Create the diskless and dataless client resources in subdirectories of the **/export/dd_resource** directory. Not all resources are required. Create only the resources to be used in your environment.

    To create the `root` resource named `root1` (required for diskless and dataless clients), enter:

```
nim -o define -t root -a server=master \
-a location=/export/dd_resource/root1 root1
```

    To create the `dump` resource named `dump1` (required for diskless and dataless clients), enter:

```
nim -o define -t dump -a server=master \
-a location=/export/dd_resource/dump1 dump1
```

    To create the `paging` resource named `paging1` (required for diskless clients), enter:

```
nim -o define -t paging -a server=master \
-a location=/export/dd_resource/paging1 paging1
```

    To create the `home` resource named `home1` (optional), enter:

```
nim -o define -t home -a server=master \
-a location=/export/dd_resource/home1 home1
```

    To create the `shared_home` resource named `shared_home1` (optional), enter:

```
nim -o define -t shared_home -a server=master
\
-a location=/export/dd_resource/shared_home1 shared_home1
```

    To create the `tmp` resource named `tmp1` (optional), enter:

```
nim -o define -t tmp -a server=master \
-a location=/export/dd_resource/tmp1 tmp1
```

    Notes:

    1. The file systems created for the NIM resources are not required, but they can be beneficial for storage management.
    2. For more information about NIM resources, see “NIM Resources” on page 83.

Adding a Diskless or Dataless Client to the NIM Environment

This procedure describes how to add diskless and dataless clients to the NIM environment by adding an entry for the client to the NIM database on the master. This provides NIM with the information required to satisfy boot requests from the client. However, resources for the diskless or dataless client machine must be initialized before the client will be able to successfully boot and configure. See “Initializing and Booting a Diskless or Dataless Machine” on page 29 for more information. Diskless clients must mount all file systems from remote servers. Dataless clients can have paging space, as well as the **/tmp** and **/home** file systems on the local disk. Neither diskless nor dataless clients have a boot image on the local disk. Therefore, they must boot over the network.

Prerequisites

1. The NIM master must be configured, and the resources for diskless or dataless clients must be defined. For more information, see “Configuring the NIM Master and Creating Resources to Support Diskless and Dataless Clients Only” on page 24.
2. You must know the subnet mask, the default gateway for the client machine, and the default gateway for the NIM master.

From Web-based System Manager

1. In the Machines container, from the Machines menu, select **New —> OK**. The Add New Machine wizard displays.
2. Follow the wizard instructions to add a diskless or dataless client to the NIM environment.

From SMIT

1. To define a diskless or dataless client, enter the **smit nim_mkmac** fast path.
2. Specify the host name of the machine.
3. The SMIT screen displayed next depends on whether NIM already has information about the client's network. Supply the values for the required fields or accept the defaults. Use the help information and the LIST option to help you specify the correct values to define the client machine.

   Note: Prior to AIX 4.2, specific network objects must be defined in addition to the steps provided in this procedure. In this procedure, NIM networks are added automatically when needed.

From the Command Line

To define a diskless or dataless client, enter:

```
nim -o define -t Diskless/Dataless \
-a platform=PlatformType -a netboot_kernel=NetbootKernelType \
-a if1=InterfaceDescription -a net_definition=DefinitionName \
-a ring_speed1=Speedvalue -a cable_type1=TypeValue \
-a iplrom_emu=DeviceName MachineName
```

Note: For detailed attribute information, see the descriptions of diskless and dataless clients in "NIM Machines" on page 73.

Example 1:

To add the diskless client with the host name `diskless1` to the NIM environment with the following configuration:

```
host name=diskless1
platform=rspc
kernel=up
network type=ethernet
subnet mask=255.255.240.0
default gateway=gw1
default gateway used by NIM master=gw_master
cable type=bnc
network boot capability=yes (no emulation needed)
```

enter the following command sequence:

```
nim -o define -t diskless -a
platform="rspc" \
-a netboot_kernel="up" -a if1="find_net diskless1 0" \
-a cable_type1="bnc" \
-a net_definition="ent 255.255.240.0 gw1 gw_master" \
diskless1
```

Example 2:

To add the dataless client with the host name `dataless1` to the NIM environment with the following configuration:

```
host name=dataless1
platform=rs6k
netboot_kernel=up
network type=token ring
subnet mask=255.255.225.0
default gateway=gw2
```

```
default gateway used by NIM master=gw_master
ring speed=16
network boot capability=no (use emulation on a diskette)
```

enter the following command sequence:

```
nim -o define -t dataless -a
platform="rs6k" \
-a netboot_kernel="up"  -a if1="find_net dataless1 0" \
-a ring_speed1="16" \
-a net_definition="tok 255.255.225.0 gw2 gw_master" \
-a iplrom_emu="/dev/fd0" dataless1
```

Notes:

1. Prior to AIX 4.2, specific network objects must be defined in addition to the steps provided in this procedure. In this procedure, NIM networks are added automatically when needed.
2. If the **find_net** keyword in the **if** attribute causes NIM to successfully match a network definition to the client definition, the **net_definition** attribute is ignored.

Initializing and Booting a Diskless or Dataless Machine

This procedure describes how to use NIM to configure and boot a machine as a diskless or dataless client in the NIM environment.

Prerequisites

1. The NIM master must be configured, and the resources for diskless and dataless clients must be defined. See “Configuring the NIM Master and Creating Resources to Support Diskless and Dataless Clients Only” on page 24.
2. The NIM client must already exist in the NIM environment. To add the client to the NIM environment, use the “Adding a Diskless or Dataless Client to the NIM Environment” on page 27 procedure.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select the diskless or dataless machine you want to initialize.
3. From the Selected menu, choose **Initialize Machine Resources**.
4. Use the dialog to specify or select the resources to use for initialization. You will specify either the Home resource or Shared Home resource for the machine, but not both.
5. After completion of the initialization operation, use the “Booting a Machine Over the Network” on page 159 procedure to boot the client machine over the network.

   Note: On older model **rspc** systems, it may be necessary to permanently set the bootlist from the firmware menus to make the client always boot over the network. For other systems, the bootlist is automatically set the first time the machine is booted as a diskless/dataless client.
6. After the client boots over the network and performs some initialization, the client will display instructions for you to select the console for the machine.

From SMIT

1. On the NIM master, enter the **smit nim_dd_init** fast path.
2. Select the client to be initialized from the list of clients displayed on your screen.
3. Supply the values for the required fields. Use the help information and the LIST option to help you specify the correct values for the initialization options.
4. After completion of the initialization operation, use the “Booting a Machine Over the Network” on page 159 procedure to boot the client machine over the network.

Note: On older model **rspc** systems, it may be necessary to permanently set the bootlist from the firmware menus to make the client always boot over the network. For other systems, the bootlist is automatically set the first time the machine is booted as a diskless/dataless client.

5. After the client boots over the network and performs some initialization, the client will display instructions for you to select the console for the machine.

From the Command Line

1. To initialize the client resources for diskless clients, enter the following on the NIM master:

```
nim -o dkls_init -a spot=SPOTName -a root=RootName \
-a dump=DumpName -a paging=PagingName ClientName
```

2. To initialize the client resources for dataless clients, enter the following on the NIM master:

```
nim -o dtls_init -a spot=SPOTName -a root=RootName \
-a dump=DumpName ClientName
```

Note: For detailed information about other attributes you can specify for the **dkls_init** and **dtls_init** operations, see “dkls_init” on page 112 and “dtls_init” on page 113.

3. After completion of the initialization operation, use the “Booting a Machine Over the Network” on page 159 procedure to boot the client machine over the network.

Note: On older model **rspc** systems, it may be necessary to permanently set the bootlist from the firmware menus to make the client always boot over the network. For other systems, the bootlist is automatically set the first time the machine is booted as a diskless/dataless client.

4. After the client boots over the network and performs some initialization, the client will display instructions for you to select the console for the machine.

Uninitializing Diskless and Dataless Machines

Diskless and dataless machines are uninitialized by performing the **reset** operation. This action also provides the option to deallocate all resources for the machine. Deallocating all resources from the diskless or dataless machine removes all root data for the machine. Without deallocating resources, the uninitialize operation deallocates just the network boot image.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select the diskless or dataless machine you want to initialize.
3. From the Selected menu, choose **Uninitialize Machine Resources**.
4. Use the dialog to uninitialize and, if desired, deallocate all resources from the client.

From SMIT

1. To uninitialize diskless and dataless machines, enter the **smit nim_dd_uninit** fast path.
2. Select the Target.
3. If you want to remove all root data, change the DEALLOCATE Resources field to **yes**.

From the Command Line

1. To uninitialize the client machine, enter the following on the NIM master:

```
nim -F -o reset ClientName
```

2. To deallocate all resources and remove root data, enter the following on the NIM master:

```
nim -o deallocate -a subclass=all
ClientName
```

Installing to an Alternate Disk on a NIM Client (cloning or mksysb)

NIM allows you to install an AIX 4.3 **mksysb** image (mksysb resource) on a NIM client's alternate disk or to clone a NIM client's current disk onto an alternate disk and apply updates. Because the client system is running during installation, less time is required than with a normal install.

> **Note:** See "alt_disk_install" on page 102 for detailed information about the different ways NIM can customize an alternate disk installation.

Prerequisites

1. The NIM master must be configured. To install a **mksysb** image onto the alternate disk, the **mksysb** resource must be defined. See "Configuring the NIM Master and Creating Basic Installation Resources" on page 7.
2. The NIM client must already exist in the NIM environment and must be running. To add the client to the NIM environment, see "Adding a Standalone NIM Client to the NIM Environment" on page 10.
3. The **bos.alt_disk_install.rte** fileset must be installed on the NIM client. To install a new fileset on a NIM Client, see "Customizing NIM Clients and SPOT Resources" on page 23.

From Web-based System Manager

1. Select the Machines Container.
2. In the contents area, select the standalone machine for the alternate disk installation.
3. From the Selected menu, choose **Alternate Disk Installation —> Clone the Rootvg to an Alternate Disk** or **Install Mksysb on an Alternate Disk**.
4. Use the dialog to finish the installation.

From SMIT

1. Enter the **smit nim_alt_mksysb** fast path from the NIM master.
2. Select the Target Machine or Target Group to Install.
3. Enter the Target Disk or Disks on the Target machine.
4. Accept the default installation options, or supply different ones in the displayed dialog fields. Use the help information and the LIST option for guidance.
5. The alternate disk installation will be initiated on the client, and progress can be seen with the **lsnim** command (**smit lsnim**). If the ″Reboot when complete?″ option is set to **yes** and the ″Phase to execute″ is **all** or includes Phase 3, the client will reboot from the newly installed disk when the **alt_disk_install** command is complete.
6. To clone a disk onto a NIM client's alternate disk, enter the **smit nim_alt_clone** fast path from the NIM master.

From the Command Line

The **alt_disk_install** command is initiated on the target system, and progress is shown with the **lsnim** command. In addition, a log kept on the target system, **/var/adm/ras/alt_disk_inst.log**, contains progress messages and any error or warning messages that might occur. The **/var/adm/ras/nim.alt_disk_install** log will contain debug information, if requested.

Installing mksysb on an Alternate Disk

Initiate the **alt_disk_install** operation by entering:

```
nim -o alt_disk_install -a source=mksysb -a mksysb=Mksysb \-a disk='diskname(s)' ClientName
```

Specify the **mksysb** resource to be used and any additional options for customizing the installation. To perform a simple alternate disk **mksysb** install, specify the **source**, **mksysb**, and **disk** resources.

Note: For detailed information about the mksysb resources, see “mksysb Resource” on page 91.

Cloning the rootvg to an Alternate Disk

To clone a disk onto a NIM client’s alternate disk, enter:

```
nim -o alt_disk_install -a source=rootvg -a disk=diskname(s) ClientName
```

Specify any additional options for customizing the installation.

Examples

The client machine `machine1` is a running system with a disk, `hdisk2`, that is not currently occupied by a volume group.

- To install this disk with a **mksysb** resource named `43mksysb` enter:

  ```
  nim -o alt_disk_install -a source=mksysb -a mksysb=43mksysb \-a disk=hdisk2 machine1
  ```

- To clone the rootvg to `hdisk2` enter:

  ```
  nim -o alt_disk_install -a source=rootvg -a disk=hdisk2 machine1
  ```

Chapter 4. NIM Client Operations

Performing NIM Client Operations

NIM operations are usually performed by an administrator on the NIM master. However, it is also possible for an administrator to perform NIM installation operations from client machines. Operations initiated from client machines are referred to as *pull installations*, because the software is downloaded from a remote server to the local machine. To reinstall the Base Operating System (BOS), select the Install BOS wizard from the Machines container. Deinstalling software and committing and rejecting updates are performed on client machines through local operations and are not supported through NIM client operations.

NIM client operations are supported through the Web-based System Manager Software application, as well as from SMIT and the command line. Because the client interfaces to NIM are very similar to the interfaces on the NIM master, they will be discussed only briefly here. Online contextual help is available for both the Web-based System Manager and SMIT interfaces. For more information on the command line interface, see the **nimclient** Command.

From Web-based System Manager

1. To start the Web-based System Manager Software application, enter **wsm**.
2. To perform NIM client operations, select and expand the NIM container.
3. Select the Machines container.
4. In the contents area, click on the machine on which you want to perform the operation.
5. From the Selected menu, choose the operation you want to perform.

From SMIT

To perform NIM client operations, enter the **smit nim** fast path.

For normal installation operations, select the **Install and Update Software** option. Submenus guide you through performing the installation.

Advanced NIM users can select one of the other main options to perform more specialized NIM operations. These options are discussed in detail in the online helps and elsewhere in this guide.

From the Command Line

To perform NIM operations on a running, configured NIM client, use the **nimclient** command. The syntax for using the **nimclient** command is nearly identical to the syntax for using the **nim** command. The only difference is that you do not need to specify the target, since it is assumed to be the client where the command is running. For more information, see the nimclient Command or the nim Command .

Example:

On the NIM master, to install the `bos.sysmgt.sysbr` fileset on the client, `client1`, using the **lpp_source**, `lpp_source1`, enter:

```
nim -o cust -a lpp_source=lpp_source1 \
-a filesets="bos.sysmgt.sysbr client1
```

The equivalent command from the client `client1` would be:

```
nimclient -o cust -a lpp_source=lpp_source1 \
-a filesets="bos.sysmgt.sysbr"
```

Chapter 5. Advanced NIM Installation Tasks

This chapter describes the following procedures for performing advanced NIM installation tasks using the Web-based System Manager application or the System Management Interface Tool (SMIT), as well as from the command line:

Controlling the Master or Client

In the NIM environment, control is held by the NIM master or the standalone client. The system allocating the resources has control. The allocation of resources is the act of making resources available to clients for NIM operations. Normally, resources are allocated automatically as part of an operation, but they may also be allocated prior to the initiation of an operation. The control status acts like a locking mechanism and remains with the client or the master until the resources are deallocated. Using NIM, if the installation of a standalone client completes successfully, the resources are automatically deallocated.

When there are no resources allocated to the standalone client by the NIM master, the standalone client takes control by allocating resources or disabling the NIM master's push permissions. The **control** attribute is managed by the master and indicates whether the master or the standalone client has permission to perform operations on the standalone client.

There are four control states indicated by the **control** attribute. You can display the **control** attribute from a NIM client by entering:

```
nimclient -l -l
StandAloneClientName
```

The **control** attribute can be displayed from the NIM master by entering:

```
lsnim -l StandaloneClientName
```

The states are:

control attribute is not set
If the **control** attribute is not displayed when listing the machine object attributes, then neither the master nor the standalone client has control.

control = master
The master has allocated resources to the client and is ready to initiate an operation (or has already initiated an operation).

control = *StandaloneClientName*	The standalone client has allocated resources and can now initiate NIM operations on itself.
control = *StandaloneClientName* **push_off**	The standalone client has prohibited the NIM master from allocating resources or initiating operations on the client. The client itself can still control the allocation of NIM resources and the initiation of NIM operations.

Disabling Master Push Permissions

From Web-based System Manager

1. From the main Web-based System Manager container, select the Software icon.
2. From the Software menu, select **NIM Client** > **Permissions**.
3. Select whether to grant or deny permission for the NIM master to initiate push installations.

From SMIT

To disable the master's push permissions, enter the **smit nim_perms** fast path from the client machine.

From the Command Line

To set **control** on the client to **push_off**, enter the following on the client machine:

```
nimclient -P
```

To re-enable push permission on the client, enter the following on the client machine:

```
nimclient -p
```

Resetting Machines

The operations performed using NIM can be very complex. To help ensure that the operations can be completed successfully, NIM requires that a machine be in the **ready** state before operations can be run on it. While an operation is being performed, the state of the machine will reflect the current operation. After the operation completes, the machine returns to the **ready** state.

If an operation on a machine is interrupted, the machine state may continue to reflect the operation. If this occurs, the machine must be reset to the **ready** state before performing any further operations. To return a machine to the **ready** state, use the NIM **reset** operation.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select a target standalone, diskless, or dataless machine to reset.
3. From the Selected menu, choose **Administration** —> **Reset NIM State**.
4. Use the dialog to reset the state of the machine.

You can also do this task from Troubleshooting. From the Selected menu, choose **Troubleshooting** —> **Clean Up Failed or Interrupted Installation**.

From SMIT

1. To return a machine to the **ready** state, enter the **smit nim_mac_op** fast path.
2. Select the target machine for the operation.
3. Select **reset** as the Operation to Perform.
4. To deallocate resources, change the `Deallocate All Resources?` field to **yes**.
5. Change the Force field to **yes**.

From the Command Line

1. To return a machine to the **ready** state, enter:

   ```
   nim -Fo reset MachineName
   ```

2. To deallocate resources, enter:

   ```
   nim -o deallocate -a
   ResourceType=ResourceName MachineName
   ```

   where `ResourceType` is the type of the resource being deallocated (for example, **lpp_source**, **SPOT**, **Script**, etc.), `ResourceName` is the name of the resource being deallocated, and `MachineName` is the name of the machine that has been allocated the resources.

   Note: Resetting a machine will not automatically deallocate all the resources that were allocated for the operation. To deallocate resources, use the NIM **deallocate** operation.

Using Client Machines as Resource Servers

Any machine in the NIM environment can be a resource server. In simple environments, the NIM master is usually used to serve all the NIM resources.

Defining resources on client machines can be beneficial for the following reasons:

- Disk space limitations on the NIM master may prohibit the storage of all the resources on a single machine.
- Resource usage may be heavy, and communications and data access bottlenecks could occur if all the resources were served by a single machine.

  For example, if you use NIM to install 200 machines on 5 different subnets, you could have a set of resources created and available on each subnet. Each set of resources would be used to install the machines on the same subnet. In addition to distributing the workload among several resource servers, this would also reduce the network traffic across the gateways between the different subnets.

From Web-based System Manager

1. Select the Resources container.
2. From the Resources menu, select **New Resources**.
3. Follow the wizard instructions to create the resource.

From SMIT

1. To create a resource on a NIM client, enter the **smit nim_mkres** fast path.
2. Select the Resource Type.
3. In the displayed dialog fields, supply the correct values for the resource options. Be sure to specify the name of the client machine for the Server of the Resource field. Use the help information or the LIST option to help you. All attributes specified when the resource is defined (such as **location** and **source**) must be local to the server machine.

From the Command Line

To create a resource on a NIM client, specify the client's NIM name for the **server** attribute when defining the resource.

Example:

To create an **lpp_source** resource named `images2` from a CD on the NIM client machine, `client_mac1`, in the `/resources/images` directory, enter:

```
nim -o define -t lpp_source -a
server=client_mac1 \
-a location=/resources/images -a source=/dev/cd0 images2
```

Defining a Machine Group

Machine groups can be defined to collect multiple clients in a common target for NIM operations. Groups can be defined for standalone, diskless, or dataless clients; but a group can only contain clients of a single type.

Web-based System Manager supports two types of machine groups. First, a temporary machine group is created when multiple machines are selected in the NIM container and an action from the Selected menu is performed. The temporary group is removed when the action is done. If you want a more permanent machine group, you can create it using the **New Machine Group** menu option in the NIM menu.

In the Web-based System Manager NIM application, machine groups are not explicitly created and managed, but ad hoc groupings are supported by multi-selecting the icons representing machines in the NIM container. Once selected, a group of machines can be administered by selecting an action from the Selected menu.

Note: You can only perform most operations on multi-selected machines of the same type.

Web-based System Manager

1. Select the Groups container.
2. From the Groups menu, select **New —> Group**
3. Select the machine type.
4. Select a machine from the list on the right, and click on the **<** button to add the machine to the new group. Continue with this step until all the desired members of the machine group are in the **Members**list on the left.
5. Click on **OK**.

From SMIT

1. To define a machine group, enter the **smit nim_mkgrp** fast path.
2. Select the type of group you want to define.
3. Enter the name of the group and member information.

From the Command Line

To define a machine group, enter:

```
nim -o define -t mac_group -a add_member=MemberName GroupName
```

For example, to create a machine group named `MacGrp1` containing previously defined machines `Standalone1`, `Standalone2`, and `Standalone3`, enter:

```
nim -o define -t mac_group -a add_member=Standalone1 \
-a add_member=Standalone2 -a add_member=Standalone3 \
-a comments="Machines for Department d03" MacGrp1
```

Adding New Members to Machine Groups

New members can be added to machine groups, however, the new member must be of the same machine type as existing members. Members can be added to machine groups using the Web-based System Manager NIM application.

From Web-based System Manager

1. Select the Groups container.
2. In the contents area, select a group.
3. From the Selected menu, choose **Add/Remove Members...**.
4. Select a machine from the list on the right, and click on the **<** button to add the machine to the new group. Continue with this step until all the desired members of the machine group are in the list on the left.
5. Click on **OK**.

From SMIT

1. To add members to a machine group, enter the **smit nim_chgrp** fast path.
2. Select the machine group to modify.
3. Specify members to add to the group. Use the LIST option to select members to add.

From the Command Line

To add a member to a machine group, enter:

```
nim -o change -a add_member=MachineName GroupName
```

For example, to add the diskless client, `diskless5`, to the machine group, `diskless_grp`, enter the following command:

```
nim -o change -a add_member=diskless5 diskless_grp
```

Alternatively, you could have specified group members in both the **define** and **change** operations by using sequenced member attributes, such as `-a member1=Standalone1 -a member2=Standalone2` and so forth.

Removing Members from Machine Groups

Members can be removed from machine groups. Whenever the last member of a machine group is removed, the group definition is also removed.

The Web-based System Manager NIM application can be used to remove members from machine groups.

From Web-based System Manager

1. Select the Groups container.
2. From the Selected menu, choose **Add/Remove Members**.
3. Select a machine from the list on the left and click on the **>** button to add the machine to the list on the right. Continue with this step until all the desired members of the machine group have been removed.
4. Click on **OK**.

From SMIT

1. To remove members from a machine group, enter the **smit nim_chgrp** fast path.
2. Select the machine group to modify.
3. Specify members to remove from the group. Use the LIST option to select members to remove.

From the Command Line

To remove a member from a machine group, enter the following command:

```
nim -o change -a rm_member=MachineName GroupName
```

For example, to remove machine, Standalone2, and add machine, Standalone4, to the group, MacGrp1, enter:

```
nim -o change -a rm_member=Standalone2 \
-a add_member=Standalone4 MacGrp1
```

Including and Excluding a Group Member from Operations on the Group

Group members may be included or exclude by using the Web-based System Manager NIM application, SMIT, or from the command line. Use the **select** operation from the command line to indicate that specific members of a machine group should be included or excluded from operations on that group. This capability is useful if an operation needs to be tried again on specific group members that failed during an initial operation on the group. When a group member is marked as being excluded, it remains so until it is included again.

From Web-based System Manager

1. Select the Groups container.
2. In the contents area, expand a group container to view the members included in that group.
3. Select a machine from those listed in the container.
4. From the Selected menu, choose **Properties**.

From SMIT

1. To include or exclude a group member from operations on the group, enter the **smit nim_grp_select** fast path.
2. Select the name of the group from which you want to include or exclude members.
3. Select the members to include or exclude.

From the Command Line

To include or exclude a group member, enter the following:

```
nim -o select -a include_all=Value -a exclude_all=Value \
-a include=MemberName -a exclude=MemberName GroupName
```

As an example, to exclude the machine, Standalone2, from further operations on machine group, MacGrp1 and to include a previously excluded machine, Standalone3, enter:

```
nim -o select -a exclude=Standalone2 -a include=Standalone3 \
MacGrp1
```

The special attributes **include_all** and **exclude_all**, when assigned a value of **yes**, can be used respectively to include or exclude all members in a group. The **select** operation evaluates command line attributes from left to right. The following example shows how to exclude all members except Standalone2 from subsequent operations on the machine group, MacGrp1:

```
nim -o select -a exclude_all=yes -a include=Standalone2 MacGrp1
```

Using the special **-g** option shows the excluded status of the group's members:

```
lsnim -g MacGrp1
```

Group member information similar to the following is displayed:

```
MacGrp1:
type   = mac_group
member1=Standalone1;ready for a NIM operation,not running;EXCLUDED
member2=Standalone2;ready for a NIM operation; currently running;
member3=Standalone3;ready for a NIM operation,not running;EXCLUDED
```

Using Resource Groups to Allocate Related Resources to NIM Machines

NIM resource groups allow association of resources so they can be allocated as a logical unit to machines prior to other NIM operations. Resource groups can only contain one of each resource type, except for **script** and **installp_bundle** resources, which may occur multiple times in a given resource group.

Resource groups are not currently supported in the Web-based System Manager application.

Defining a Resource Group

From SMIT

1. To define a resource group, enter the **smit nim_mkgrp_resource** fast path.
2. Enter the name of the group with member information.

From the Command Line

To define a resource group, enter:

```
nim -o define -t res_group -a
ResourceType=ResourceName GroupName
```

As an example, to create a resource group named `ResGrp1` containing previously defined resources, `images1`, `spot1`, `bosinst_data1`, and `bundle1`, enter:

```
nim -o define -t res_group -a
lpp_source=images1 -a spot=spot1 \
-a bosinst_data=bosinst_data1 -a installp_bundle=bundle1 \
-a comments="BOS Install Resources" ResGrp1
```

Allocating a Resource Group

From SMIT

1. To allocate a resource group, enter the **smit nim_alloc** fast path.
2. Select the machine or machine group from the list of defined machines (for example, `Standalone1`).
3. A list of resource groups is displayed. Select the resource group you want to allocate.

From the Command Line

To allocate a resource group, enter:

```
nim -o allocate -a group=ResGroupName
TargetName
```

For example, to allocate a resource group named `ResGrp1` to a machine named `Standalone1`, enter:

```
nim -o allocate -a group=ResGrp1
Standalone1
```

Alternatively, the group resource can be specified on the command line to the operation. For example, to allocate the resource group, `ddResGrp`, while performing the **dkls_init** operation on a group of diskless machines named `DklsMacs`, enter:

```
nim -o dkls_init -a group=ddResGrp
DklsMacs
```

Defining Default Resource Groups

After a resource group is defined, you may want to specify the group as the set of defaults for all operations that require resources. Set the **default_res** attribute on the master to the name of the resource group that you want to be the default.

From SMIT

1. To define default resource groups, enter the **smit nim_grp** fast path.
2. Choose Select/Unselect a Default Resource Group.
3. Fill in the name of the group that is to act as the default.

From the Command Line

To define default resource groups, enter:

```
nim -o change -a default_res=ResGroupName
master
```

For example, if the resource group, `ResGrp1`, should be the set of default resources for all NIM operations, enter:

```
nim -o change -a default_res=ResGrp1
master
```

> **Note:** All applicable resources are allocated from the group specified as the default for all operations, except for **installp_bundle** for a **maint** operation.
>
> A resource from the default group will only be allocated if a resource of the same type is not already allocated and if a resource of that type is not specified on the command line for automatic allocation. The exceptions are the **script** and **installp_bundle** resources, of which all occurrences in the resource group and specified on the command line will be allocated.
>
> Default members can be overridden by specifying a null value in the attribute assignment for that resource.

The following **bos_inst** operation allocates all applicable **bos_inst** resources from the resource group specified as the default, except for the **bosinst_data** resource:

```
nim -o bos_inst -a bosinst_data=
Standalone1
```

Managing Software on Standalone Clients and SPOT Resources

The commands for managing software on standalone clients and **SPOT** resources are generally the same. Specify the name of the machine, group, or **SPOT** as the target of the option.

> **Note:** If the **SPOT** is currently allocated to a NIM client, NIM prevents the change to the **SPOT**. Use the **Force** (**-F**) option to force the operation.

Software updates to a **SPOT** cause the **SPOT**'s network boot images to be rebuilt when necessary. If you think the boot images are bad, you can force them to be rebuilt using the NIM **check** operation.

Software updates to a **SPOT** may also cause software updates to occur in the root parts of diskless and dataless clients of the **SPOT**. This will occur automatically. You can force a synchronization of the client root parts using the NIM **sync_roots** operation on the **SPOT**.

For information on how to install additional software on standalone clients and SPOT resources, see "Customizing NIM Clients and SPOT Resources" on page 23.

Listing Software Installed on a Standalone Client or SPOT

From Web-based System Manager

1. Select the Machines container.

2. In the contents area, select a target machine (master or standalone), or in the Resources container, select a target **SPOT** resource.
3. From the Selected menu, choose **List Installed Software** —> **All Installed**.

From SMIT

1. To list software installed on a standalone client or **SPOT**, enter the **smit nim_list_installed** fast path.
2. Select the menu item that describes the list operation you want to perform.
3. Select a target for the operation.
4. In the displayed dialog fields, supply the required values. Use the help information or the LIST option to help you.

From the Command Line

Enter the following command:

```
nim -o lslpp [-a lslpp_flags=LslppFlags]
TargetName
```

where *lslppFlags* are the flags to be passed to the **lslpp** command, and *TargetName* is the name of the client or **SPOT** object.

For example:

```
nim -o lslpp -a lslpp_flags=La spot1
```

Listing Software Updates, Installed on a Standalone Client or SPOT, by Keyword

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select a target machine (master or standalone), or in the Resources container, select a target **SPOT** resource.
3. From the Selected menu, choose **List Installed Software** —> **Fix (APAR) Status**.
4. Use the dialog to list the installation status of specific installed fixes.

From SMIT

1. To list fixes installed on a standalone client or **SPOT** by APAR number or keyword, enter the **smit nim_mac_op** fast path for standalone clients, or enter the **smit nim_res_op** fast path for **SPOTs**.
2. Select the standalone client or **SPOT** resource object.
3. Select the **fix_query** operation.
4. Select the desired **fix_query** flags or accept the default settings. Specify the **fix_bundle** object name; or to check the installation status of an APAR, specify the fix APAR numbers. If you leave both blank, all known fixes are displayed.

From the Command Line

Enter the following command:

```
nim -o fix_query [ -afixes="FixKeywords" ]
\
[-afix_bundle=BundleName ] [ -afix_query_flags=FixQueryFlags ] \
TargetName
```

where *FixKeywords* are APAR numbers; *FixBundleName* is the object name of the **fix_bundle** resource; *FixQueryFlags* are optional flags to the **fix_query** operation, and *TargetName* is the client, group, or **SPOT** for which to display fix information.

Valid *FixQueryFlags* are as follows:

-a Displays symptom text.
-c Displays output in colon-separated format.
-F Returns failure unless all filesets associated with a fix are installed.
-q Quiet option; if **-q** is specified, no heading is displayed.
-v Verbose option; gives information about each fileset associated with a fix (keyword).

For example:

- To query the `fix` database on `standalone1` to determine if all fileset updates for fix `IX12345` are installed, enter:

```
nim -o fix_query -afixes=IX12345
standalone1
```

- To list fix information for all known fixes installed on `spot1`, with symptom text, enter:

```
nim -o fix_query -afix_query_flags=a
spot1
```

Maintaining Software on Standalone Clients and SPOT Resources

NIM uses the **installp** command to construct a **SPOT** by installing in the **SPOT** the software products that each **SPOT** needs to support the NIM environment. Because the **installp** command also supports software maintenance tasks, you can perform these tasks on **SPOTs** as well. For example, you can remove previously installed optional software from a **SPOT** when they are no longer being used. This kind of task is accomplished by performing the NIM **maint** operation on a **SPOT** using the Web-based System Manager NIM application, SMIT, or command line interface. You interact with the **installp** command by supplying the **installp_flags,** and either **filesets** or **installp_bundle** attributes.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select a target standalone machine, or in the Resources container, select a target **SPOT** resource.
3. From the Selected menu, choose **Software Utilities** —> **Commit Applied Updates**, **Reject Applied Updates**, or **Remove Software**, depending upon the task you want to perform.

From SMIT

1. To perform software maintenance, enter the **smit nim_task_maint** fast path.
2. Select the menu item that describes the maintenance that you want to perform.
3. Select the target for the operation.
4. In the displayed dialog fields, supply the required values. Use the help information or the LIST option to help you.

From the Command Line

Enter the following command:

```
nim -o maint -a
installp_flags="InstallpFlags"  \
[-a filesets="FileSetNames" | \
-a installp_bundle=BundleResourceName ] [-F] TargetName
```

where *InstallpFlags* are the flags you want to pass to the **installp** command; *FileSetNames* are the names of the filesets or packages you want to maintain; *BundleResourceName* is the object name of the **installp_bundle** resource; and *TargetName* is the object name of the standalone client, group, or **SPOT**.

For example:

- To remove the `bos.adt` and `bos.INed` software packages from `standalone1`, enter:

```
nim -o maint -a filesets="bos.adt bos.INed" -a
\
installp_flags="-u" standalone1
```

- To remove the `bos.INed` software package from `spot1`, which is allocated to diskless or dataless clients, without deallocating `spot1` first, enter:

```
nim -o maint -F -a filesets=bos.INed -a
installp_flags="-u" \
spot1
```

- To remove the packages from `spot1` which are listed in the bundle pointed to by the **installp_bundle** resource object, `bundle1`, enter:

```
nim -o maint -a installp_flags="-u" -a
installp_bundle=bundle1 \
spot1
```

- To clean up from an interrupted software installation on `spot1`, enter:

```
nim -o maint -a installp_flags="-C"
spot1
```

Rebuilding Network Boot Images for a SPOT

From Web-based System Manager

1. Select the Resources container.
2. In the contents area, select a target **SPOT**.
3. From the Selected menu, choose **Check SPOT**.
4. Use the dialog to select the Build debug network boot images and/or the force option, if needed.

You can also perform this task from Troubleshooting. From the Selected menu, choose **Troubleshooting —> Build Non-Debug Network Boot Images**.

From SMIT

1. To rebuild network boot images for a **SPOT**, enter the **smit nim_res_op** fast path.
2. Select the **SPOT**.
3. Select the **check** operation.
4. In the displayed dialog fields, set the Force option to **yes**.

From the Command Line

To force the rebuild of the boot images, enter:

```
nim -Fo check SPOTName
```

For information on how to install additional software on standalone clients and SPOT resources, see “Customizing NIM Clients and SPOT Resources” on page 23.

Maintaining Software in an lpp_source

To add or remove software in an **lpp_source**, you simply add or remove the installation image from the **lpp_source** directory, and then initiate the NIM **check** operation on the **lpp_source**.

Copying Software to an lpp_source

From Web-based System Manager

1. Select the Resources container.
2. In the contents area, select an **lpp_source**.

3. From the Selected menu, choose **Properties**. The General page of the properties notebook displays.
4. From the General page, identify the location of the resource. Close the notebook.
5. From the Resources menu, select **Copy Software to Directory**, and specify as the destination directory, the location of the resource identified in the notebook.
6. After the copy is completed, select the **lpp_source** and from the Selected menu, choose **Check NIM State**. This action updates the table of contents (**.toc**) file for the **lpp_source**.

From SMIT

1. To copy software from installation media to an **lpp_source**, insert the installation media in the appropriate drive of the **lpp_source** server.
2. To copy the software to the **lpp_source** directory, enter **smit bffcreate** from the resource server.
3. Enter the INPUT device / directory for software.
4. In the displayed dialog fields, supply the correct values or accept the default values. Be sure to specify the **lpp_source** location for the directory to store the installation images. Use the help information and the LIST option to help you.

From the Command Line

1. Copy the software from the media to the **lpp_source** directory.
2. Perform the NIM check operation on the **lpp_source** by entering the following command:

```
nim -o check Lpp_sourceName
```

Removing Software from an lpp_source

To remove software from an **lpp_source**, delete the installation image from the **lpp_source** directory.

Note: This function is only available from the command line interface.

From the Command Line

1. Remove the installation image from the **lpp_source** directory.
2. Perform the NIM check operation on the **lpp_source** by entering the following command:

```
nim -o check Lpp_sourceName
```

Running the NIM check Operation

After adding or removing software, 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 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 System images on a machine.

From Web-based System Manager

1. Select the Resources container.
2. In the contents area, select a target **lpp_source** resource.
3. From the Selected menu, choose **Check NIM State**.

From SMIT

1. To run the NIM **check** operation, enter the **smit nim_res_op** fast path.
2. Select the **lpp_source** for the operation.
3. Select **check** for the operation to be performed.

From the Command Line

To initiate the NIM **check** operation on the **lpp_source**, enter:

```
nim -o check Lpp_sourceName
```

If the **lpp_source** is currently allocated to a client, use the **Force** option as follows:

```
nim -F -o check Lpp_sourceName
```

Viewing Installation, Configuration, and Boot Logs

After installing a standalone machine, use the **showlog** operation to check the installation results by viewing the installation, boot, and configuration logs. One of several log types can be viewed by specifying one of the following as the value of the **log_type** attribute to the **showlog** operation:

devinst	Output from the installation of key system and device-driver software
niminst	Output from the installation of user-specified software (including installation of NIM client software during a **bos_inst** operation)
bosinst	Output from the BOS installation program
boot	The machine's boot log
lppchk	A log of the output from the **lppchk** operation executed on a standalone NIM client
script	Output from any configuration script resources allocated for a **bos_inst** operation
nimerr	Errors encountered during execution of the **nim** command.

By default, the **showlog** operation applied to a standalone machine displays the **niminst** log and shows the output logged when software was last installed on the machine using NIM. The last entry is also shown by default for the **script** and **lppchk** logs. The entire contents of the **niminst**, **script**, and **lppchk** logs can be displayed by assigning the **full_log** attribute a value of `yes` when executing the **showlog** operation. The entire log is shown for all other log types.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select a target machine (master, standalone, diskless, or dataless), or in the Resources container, select a target **SPOT**.
3. From the Selected menu, choose **Troubleshooting** —> **Show NIM Logs**.
4. Use the dialog to select the log you want to examine.

From SMIT

1. Enter the **smit nim_mac_op** fast path to view a machine's log, or enter **smit nim_res_op** to view a **SPOT**'s log.
2. Select the object name of the machine or **SPOT** whose log you want to view.
3. Select **showlog** from the list of operations.
4. Select the log type to be viewed.
5. Specify if the full log should be viewed (only applicable to **script**, **lppchk**, and **niminst** logs).

From the Command Line

To view a log on a standalone machine or **SPOT**, enter:

```
nim -o showlog -a LogType=value
ObjectName
```

where `LogType` represents the log you want to view, and `ObjectName` is the name of the machine or **SPOT** whose log will be viewed.

Verifying Installation with the lppchk Operation

You can use the **lppchk** operation to check the integrity of installed software. This is particularly useful when investigating functional problems in software.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select a target standalone machine, or in the Resources container, select a target **SPOT**.
3. From the Selected menu, choose **Troubleshooting** —> **Verify Installed Software**.
4. Use the dialog to select whether to verify all or some installed software on the selected machine or **SPOT**.

From SMIT

1. Enter the **smit nim_mac_op** fast path to check software on a machine, or enter **smit nim_res_op** to check software on a **SPOT**.
2. Select the target of the **lppchk** operation.
3. Select the desired verification mode.

From the Command Line

Enter the following command:

```
nim -o lppchk -a filesets=FilesetName \
-a lppchk_flags="lppchkFlags" ObjectName
```

where `FilesetName` is the name of a single fileset (or a name wildcarded with the * character), and `ObjectName` is the name of the machine or **SPOT** which is the target of the **lppchk** operation. Valid **lppchk_flags** are defined as follows:

-f Fast check (file existence, file length)
-c Checksum verification
-v Fileset version consistency check (default)
-l File link verification

Note: Only one of the flags **-f**, **-c**, **-v**, or **-l** may be specified.

-u Update inventory (only valid with **-c** or **-l**)
-m*n* Controls detail of messages. *n* equals1 to 3, where 3 is the most verbose.

For example, to perform the **lppchk** operation while verifying checksums for all filesets on the machine named `Standalone1`, enter the following:

```
nim -o lppchk -a lppchk_flags="-c"
Standalone1
```

Using NIM to Install Clients Configured with Kerberos Authentication

Normally, NIM relies on Standard AIX authentication to allow the NIM master to remotely execute commands. Standard AIX authentication utilizes the **.rhosts** file to provide this capability. While NIM functionality depends on its ability to remotely execute commands, some system environments require stricter authentication controls. Kerberos authentication provides a higher level of authentication for executing remote commands on the system without disabling NIM's capabilities.

Using NIM to Install Clients Configured with Kerberos 4 Authentication

In AIX 4.3.3 and later, NIM can be used to install machines in a RS/6000 SP environment configured for Kerberos 4 authentication. Clients configured for Kerberos 4 authentication will contain a **$HOME/.klogin** file for the root user. This file will determine what ticket is required to allow remote command execution. The user must obtain the required ticket before attempting to execute remote commands through NIM.

The NIM master and all secure clients must have the IBM Parallel System Support Program for AIX AIX 3.1 or greater installed and configured.

If secure clients will be reinstalled with BOS (Base Operating System), the authentication methods on the NIM master should be set for both Kerberos 4 and Standard UNIX. This is because NIM will not have configured Kerberos 4 on the client after the BOS is installed. NIM will therefore have to rely on standard rhosts to guarantee that it can remotely execute commands on the client until the client can be configured with Kerberos 4 and made into a secure client.

If only software customization and maintenance will be performed, then the NIM master must have its authentication methods set to match those of the clients. To manage secure clients, the master will need authentication methods set to include Standard UNIX.

See the *SP Administration Guide* for more information on installing and configuring Kerberos 4.

Using NIM to Install Clients Configured with Kerberos 5 Authentication

In AIX 4.3.2 and later, NIM can be used to install machines in an environment configured for Kerberos 5 authentication. Clients configured for Kerberos 5 authentication will contain a **$HOME/.k5login** file for the root user. This file will contain an entry that specifies what host token is required to allow remote command execution. This entry will follow the form:

hosts/hostname/self@cell

The NIM master and all secure clients must have DCE installed and configured at a level greater than or equal to 2.2.1.

If secure clients will be reinstalled with BOS (Base Operating System), the authentication methods on the NIM master should be set for both Kerberos 5 and Standard UNIX. This is because the client will not have DCE or Kerberos 5 configured and running after the BOS is installed. NIM will therefore have to rely on standard **rhosts** to remotely execute commands on the client until it can be configured with Kerberos 5 and made into a secure client.

If only software customization and maintenance will be performed, then the NIM master must have its authentication methods set to match those of the clients. To manage secure clients, the master will need authentication methods set to include Standard UNIX.

See the *Kerberos Version 5 Installation Guide* for more information on installing and configuring Kerberos 5.

Concurrency Control

NIM installations can become severely bogged down when they are being performed on a large number of clients at the same time. This can be caused by network bandwidth or workload on the NIM servers. Users can ease the severity of this situation by controlling the number of clients installing at the same time.

The **concurrent** and **time_limit** attributes can be used in conjunction with the **bos_inst**, **cust**, and **alt_disk_install** operations to control the number of client machines being operated on simultaneously

from a client group. The **concurrent>** attribute controls the number of clients in a group that are processing a particular operation at one time. Once a client finishes the operation, another client will initiate the operation one at a time. The **time_limit** attribute prohibits NIM from initiating an operation on any more clients of the group, after the specified time (in hours) has elapsed.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select multiple targets.
3. From the Selected menu, choose any of the following options:
   - **Install Operating System**
   - **Install/Update Software**
   - **Alternate Disk Install**
4. From any of those dialogs select the **NIM settings** or **Advanced** button.
5. In those dialogs, a section containing the Concurrency Controls can be specified.

   Note: Web-based System Manager does not provide support for continuing after a failure or if the group of machines were individually selected and the time limit expired. The user must reselect the clients that failed or were not attempted and then reissue the command.

From SMIT

The Concurrency Control attributes can be accessed from all SMIT panels under the Install and Update Software menu and the Alternate Disk Installation menu.

From the Command Line

The **concurrent** and **time_limit** attributes can be used in conjunction with the **bos_inst**, **cust** and **alt_disk_install** operations.

For example, to have the fileset **bos.games** installed on only 5 machines from the client group **tmp_grp** at one time, enter the following command:

```
nim -o cust -a lpp_source=lpp_source1 -a filesets=bos.games \
-a concurrent=5 tmp_grp
```

In this example, to BOS install only 10 clients from **tmp_grp**, using **lpp_source**, `lpp_source1`, and SPOT, `spot1`, with no other installs permitted after 3 hours have elapsed, enter the following command:

```
nim -o bos_inst -a lpp_source=lpp_source1 -a spot=spot1 \
-a concurrent=10 -a time_limit=3 tmp_grp
```

Note: The Concurrency Controlled operation can complete and leave the group in one of three states. These states are:

1. All machines install successfully.
2. Some machines may fail the install.
3. If the **time_limit** attribute was used, time may have expired before the install operation was complete.

In the first situation the group will get set back to the state prior to the operation. In the second and third situations, the group will be left in a state that indicates some machines have completed and some have not. Problems with failing machines should be investigated. At this point, the user can continue with the machines that did not complete by re-issuing the command on the group. Alternatively the user may wish to ″reset″ the group which will set the group back to its state prior to the Concurrency Controlled operation.

Chapter 6. Advanced NIM Configuration Tasks

This chapter describes the following procedures for performing advanced Network Installation Management (NIM) configuration tasks using the Web-based System Manager NIM application or the System Management Interface Tool (SMIT), as well as from the command line:

Removing Machines from the NIM Environment

Removing a machine from the NIM environment can be done by removing the client information from the NIM database.

> **Note:** When a client is removed from the NIM environment, NIM attempts to remove the **/etc/niminfo** file from the client machine. However, the client fileset and rhost permission for the NIM master must be removed manually from the client system if such additional cleanup is desired.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select a machine to remove from the NIM environment.
3. From the Selected menu, choose **Delete** to remove the machine.

From SMIT

1. To remove a machine from the NIM environment, enter the **smit nim_rmmac** fast path.
2. Select the machine to remove.
3. In the displayed dialog fields, accept the defaults.

From the Command Line

To remove a machine from the NIM environment, enter:

```
nim -o remove MachineName
```

where `MachineName` is the name of the machine to be removed.

Creating Additional Interface Attributes

The primary interface or the first interface (**if1**) is created when the master is activated, and a sequence number is used to identify the additional interfaces (**if2**, **if3,** ...) in the machine object definition. To create an additional **if** attribute for the master object, use either Web-based System Manager, SMIT, or the **nim -o change** command operation.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select any machine (master, standalone, diskless, or dataless).
3. From the Selected menu, choose **Properties**. The General page of the Machine Properties notebook for the selected machine displays.
4. Click the NIM Interfaces tab.
5. Follow the dialog instructions.

From SMIT

1. To create an additional **if** attribute, enter the **smit nim_mac_if** fast path.
2. Select the Define a Network Install Interface option.
3. Select the machine object name. In the example, this is `master`.
4. Enter the host name for the interface.
5. Complete the network-specific information in the entry fields on the Define a Network Install Interface screen.

   Note: If a NIM network does not already exist corresponding to the IP address of the host name specified for the interface, additional network information will be requested so the network can be defined.

From the Command Line

To create an additional **if** attribute for the master object, enter:

For Token-Ring:

```
nim -o change -a
ifseq_no='NetworkObjectName AdapterHostName \
AdapterHardwareAddress' -a ring_speedseq_no=Speed master
```

For Ethernet:

```
nim -o change -a
ifseq_no='NetworkObjectName AdapterHostName \
AdapterHardwareAddress' -a cable_typeseq_no=Type master
```

For FDDI:

```
nim -o change -a
ifseq_no='NetworkObjectName AdapterHostName \
AdapterHardwareAddress' master
```

For other networks:

```
nim -o change -a
ifseq_no='NetworkObjectName AdapterHostName \
AdapterHardwareAddress' master
```

Note: If you do not know the name of the NIM network to which the interface is attached or if a network corresponding to the interface has not been defined, use the **find_net** keyword and **net_definition** attribute as described in "Defining NIM Clients" on page 74.

In the example, the following command is run:

```
nim -o change -a if2='Network2 srv1_ent 0'
-a \
cable_type2=bnc master
```

With this syntax, another **if** attribute is created for the master, which tells NIM that the master has an Ethernet interface that uses a host name of `srv1_ent`, that the Ethernet adapter's hardware address is `0` (not used), and that the master connects to the `Network2` network object.

To display detailed information about the master which will now show the **if2** attribute, enter:

```
lsnim -l master
```

The command produces output similar to the following:

```
master:
  class             =  machines
  type              =  master
  Cstate            =  ready for a NIM operation
  reserved          =  yes
  platform          =  rs6k
  serves            =  boot
  serves            =  nim_script
  comments          =  machine which controls the NIM environment
  Mstate            =  currently running
  prev_state        =  ready for a NIM operation
  if1               =  Network1 server1 10005AA88399
  master_port       =  1058
  registration_port =  1059
  ring_speed1       =  16
  if2               =  Network2 Srv1_ent 02608c2e222c
  cable_type2       =  bnc
```

Defining /usr versus non-/usr SPOTs

A **SPOT** resource contains operating system files that are normally installed in the **/usr** file system of a machine. If disk space is limited on a machine or a **SPOT** must be created quickly, it may be helpful to convert the machine's **/usr** file system to a **SPOT** instead of creating an entirely separate **SPOT** at a different location.

If the **/usr** file system of a machine is converted to a **SPOT**, additional software will be installed on the machine to provide support for machines with different hardware configurations. Most of the operating system files will already be installed on the system and will not be reinstalled when the **SPOT** is created.

After a **/usr** file system is converted to a **SPOT**, all software installation and maintenance operations on the machine should be performed using NIM on the **/usr SPOT** resource that was created. This will ensure that all necessary **SPOT** operations are performed in addition to software installation/maintenance on the machine.

From Web-based System Manager

1. Select the Resources container.
2. From the Resources menu, select **New Resource**.
3. Follow the wizard instructions to create the **SPOT** resource.

From SMIT

1. To create a **/usr SPOT**, enter the **smit nim_mkres** fast path.
2. Select the Resource Type.
3. Type **/usr** in the Location of Resource field.
4. Supply the values or accept the defaults for all other fields on this screen.

From the Command Line

To create a **/usr-SPOT**, enter:

```
nim -o define -t spot -a
server=ServerName \
-a location=/usr -a source=SourceName ResourceName
```

Example:

To convert the **/usr** file system on the machine, `client1`, to a **SPOT** named `usrspot` using `lppsource1` as the source for additional installation images, enter:

```
nim -o define -t spot -a server=client1
-a location=/usr \
-a source=lpp_source1 usrspot
```

Using the installp Command

Once you convert a **/usr** file system to a **SPOT**, it is not recommended that you use the **installp** command to install or maintain software on the machine serving the **SPOT**. The diskless and dataless clients and network boot images associated with the **SPOT** will not be updated by the **installp** command unless it is invoked using NIM's **cust** or **maint** operations. If you need to use the **installp** command to install or maintain software on a **/usr SPOT** server, use the following steps:

1. Ensure that all NIM operations on the server and any clients associated with the **SPOT** are complete.
2. Deallocate the **SPOT** from all standalone clients.
3. Run the **installp** command.
4. Run the **check** operation on the **SPOT** after the **installp** command has completed:

   ```
   nim -o check -F usrSPOTName
   ```

   Note: The **-F** flag is required for rebuilding the boot images.
5. If this **SPOT** is being used to serve diskless or dataless clients, resynchronize all diskless and dataless clients with the **SPOT** after the **installp** command completes by issuing the **nim** command with the **sync_roots** operation for the **/usr SPOT**:

   ```
   nim -o sync_roots usrSPOTName

   nim -o check -F usrSPOTName
   ```

   The **cust** and **maint** operations must be used to manage software installed on non-**/usr SPOT**s.

Re-Creating SPOT Resources from Existing Directories

Defining NIM resources from existing files and directories can usually be done quickly and easily by specifying the **server** and **location** attributes to the **nim - o define** command. SPOT resources take longer to define because software must be installed from installation images into the SPOT location. Currently the **nim -o** command line interface always builds a SPOT from installation images. However, if a directory structure for a SPOT already exists from a prior creation, then it is possible to call a NIM method directly to redefine the SPOT without reinstalling all the software.

The need to define a SPOT from an existing SPOT directory typically arises only when it is necessary to rebuild the NIM database during system recovery.

To define a SPOT from a directory that previously had a SPOT installed in it, use the following command:

/usr/lpp/bos.sysmgt/nim/methods/m_mkspot -o -a server=(server) \ -a location=(location) -a source=no (spotname)

Example :

A SPOT named **spot1** was created on the NIM master in the **/export/spot directory**. Later, the NIM database became corrupted and has to be rebuilt. The SPOT files are still on the machine, but the SPOT must be redefined to NIM using the following command:

/usr/lpp/bos.sysmgt/nim/methods/m_mkspot -o -a server=master \ -a location=/export/spot -a source=no spot1

Defining an lpp_source on CD-ROM versus Disk

Normally an **lpp_source** resource is created by copying installation images from installation media to the hard disk of the **lpp_source** server. If disk space is limited on the server or if an **lpp_source** is needed quickly, it may be useful to use a directory mounted from CD-ROM installation media as the **lpp_source**.

From Web-based System Manager

1. Select the Resources container.
2. From the Resources menu, select **New Resource**.
3. Follow the wizard instructions to create the **lpp_source** resource.

Note: You can also define an **lpp_source** resource through the Configure NIM wizard, both when you are configuring your environment, and after configuration.

From SMIT

1. Mount the CD as a **CDROM** file system. The installation images can be found in the **/usr/sys/inst.images** directory under the mount point of the **CDROM** file system.
2. To define the **lpp_source** using the directory of install images, enter the **smit nim_mkres** fast path.
3. Specify the name of the machine with the CD-ROM as the Server.
4. Specify *CD_MountPoint*/**usr/sys/inst.images** as the location of the **lpp_source**, and leave the Source field blank.

From the Command Line

1. Mount the CD as a **CDROM** file system. The installation images can be found in the **/usr/sys/inst.images** directory under the mount point of the **CDROM** file system.
2. Define the **lpp_source** using the directory of install images for the **location** attribute. Do not specify a value for the **source** attribute, since an existing set of images will be used. With the CD mounted at **/cdfs** on the NIM master, to define an **lpp_source** named `cd_images`, enter:

```
nim -o define -t lpp_source -a server=master
\
-a location=/cdfs/usr/sys/inst.images cd_images
```

Establishing a Default NIM Route Between Networks

This procedure describes how to create default NIM routes for two Networks (for example, `Network1` and `Network3`).

From Web-based System Manager

1. Select the Networks container.
2. In the contents area, select any network.
3. From the Selected menu, choose **Properties**. The General page of the Properties notebook for the selected network displays.
4. Click the NIM Routes tab. The NIM Routes page of the Properties notebook displays.
5. Use the NIM Routes page to add the default route.

From SMIT

1. To create default NIM routes, enter the **smit nim_mkdroute** fast path.
2. In the displayed dialog fields, supply the values or accept the defaults. Use the help information and the LIST option to help you.

From the Command Line

To create a default NIM route for a network, enter:

```
nim -o change -a routingseq_no='default
<Gateway>' NetworkObject
```

where `default` is the reserved keyword used by NIM to indicate a default route, and `Gateway` is the host name (or IP address) of the interface that clients on `NetworkObject` use to contact other networks in the NIM environment.

For example, to establish default NIM routes for `Network1` and `Network3`, enter:

```
nim -o change -a routing1='default
gw1_tok' Network1
nim -o change -a routing1='default gw1_fddi' Network3
```

where `gw1_tok` is the host name of the default gateway for machines on `Network1`, and `gw1_fddi` is the host name of the default gateway for machines on `Network3`.

The detailed information for the network objects now shows the added default routes. To display the detailed information for the two networks, enter:

```
lsnim -l Network1 Network3
```

which produces output similar to the following:

```
Network1:
        class           = networks
        type            = tok
        net_addr        = 9.101.1.0
        snm             = 255.255.255.0
        Nstate          = ready for use
        prev_state      = ready for use
        routing1        = default gw1_tok
Network3:
        class           = networks
        type            = fddi
        net_addr        = 9.101.3.0
        snm             = 255.255.255.0
```

```
Nstate          = ready for use
prev_state      = information is missing from this
                  object's definition
routing1        = default gw1_fddi
```

Establishing a Static NIM Route Between Networks

This procedure describes how to create a static NIM route between two networks (for example, Network1 andNetwork3).

From Web-based System Manager

1. Select the Networks container.
2. In the contents area, select any network.
3. From the Selected menu, choose **Properties**. The General page of the Properties notebook for the selected network displays.
4. Click the NIM Routes tab. The NIM Routes page of the Properties notebook displays.
5. Use the NIM Routes page to add the static route.

From SMIT

1. To create a static NIM route, enter the **smit nim_mkroute** fast path.
2. In the displayed dialog fields, supply the values or accept the defaults. Use the help information and the LIST option to help you.

From the Command Line

To create a static NIM route between two networks, enter:

```
nim -o change -a
routingseq_no='DestinationNetworkObject \
Gateway1 Gateway2' NetworkObject
```

where Gateway1 is the host name of the interface that clients on NetworkObject use to get to DestinationNetworkObject, and Gateway2 is the host name that clients on DestinationNetworkObject use to get back to NetworkObject.

For example, to establish a NIM route between Network1 and Network3, enter:

```
nim -o change -a routing1='Network3
gw1_tok gw1_fddi' Network1
```

where gw1_tok is the host name of the gateway that machines on Network1 use to communicate with machines on Network3, and gw1_fddi is the host name of the gateway that machines on Network3 use to communicate with machines on Network1.

The detailed information for the network objects now shows the added routing attributes.

To display the detailed information about the two networks, enter:

```
lsnim -l Network1 Network3
```

The command produces output similar to this:

```
Network1:
   class       =  networks
   type        =  tok
   net_addr    =  9.101.1.0
```

```
    snm         =  255.255.255.0
    Nstate      =  ready for use
    prev_state  =  ready for use
    routing1    =  Network3 gw1_tok
Network3:
    class       =  networks
    type        =  fddi
    net_addr    =  9.101.3.0
    snm         =  255.255.255.0
    Nstate      =  ready for use
    prev_state  =  information is missing from this object's
                   definition
    routing1    =  Network1 gw1_fddi
```

Recovering the /etc/niminfo File

The **/etc/niminfo** file, which resides on the master and running NIM clients, is required to run NIM commands and perform NIM operations. If the **/etc/niminfo** file is accidentally deleted, you can rebuild the file.

From Web-based System Manager

1. Select the NIM container.
2. From the NIM menu, select **Advanced Configuration —> Rebuild Master Configuration File**.

> **Note:** The Configure NIM wizard will detect when you do not have a **niminfo** file, yet do have NIM database entries. The wizard queries whether it should rebuild the master configuration file.

From the Command Line

Enter the following command from the master to rebuild the file:

```
nimconfig -r
```

To rebuild the the **/etc/niminfo** file from a running NIM client, enter:

```
niminit -a master_port=PortNumber -a
master=MasterHostName \
-a name=ClientMachineObjectName
```

Backing Up the NIM Database

To back up the NIM database, you will be prompted for the name of a device or a file to which the NIM database and the **/etc/niminfo** file will be backed up. The level of the installed NIM master fileset will also be written to a file called **/etc/NIM.level** and saved in the backup. A backup of a NIM database should only be restored to a system with a NIM master fileset which is at the same level or a higher level than the level from which the backup was created.

From Web-based System Manager

1. From the NIM container, from the NIM menu, select **Back Up Database**.
2. Use the dialog to specify the backup device or file.

From SMIT

To back up the NIM database, enter the **smit nim_backup_db** fast path.

From the Command Line

Save the following NIM files:

/etc/niminfo

/etc/objrepos/nim_attr

/etc/objrepos/nim_attr.vc

/etc/objrepos/nim_object

/etc/objrepos/nim_object.vc

Restoring the NIM Database and Activating the NIM Master

> **Note:** A NIM database should only be restored to the same or later level of NIM that was used for the backup.

From Web-based System Manager

1. From the NIM container, from the NIM menu, select **Restore Database**.
2. Use the dialog to specify the restore device or file.

From SMIT

To configure a NIM master from a NIM database backup, enter the **smit nim_restore_db** fast path.

From the Command Line

Restore the files saved in the “Backing Up the NIM Database” on page 58 procedure .

Unconfiguring the NIM Master

This operation removes the NIM daemons from the system and removes all configuration from the NIM database. The NIM master should only be unconfigured if the NIM environment is to be completely redefined or if the NIM master fileset is to be removed from the system.

From Web-based System Manager

1. From the NIM Container, from the NIM menu, select **Unconfigure Environment**.
2. You have the option to back up the NIM database before starting the unconfigure action.

From SMIT

To unconfigure a NIM master, enter the **smit nim_unconfig** fast path.

The SMIT screen will prompt you to first back up your NIM database before unconfiguring the NIM master.

From the Command Line

To unconfigure a NIM master, enter **nim -o unconfig master**.

Booting Diagnostics

Hardware diagnostics can be performed on all NIM clients using a diagnostic boot image from a NIM server, rather than booting from a diagnostic tape or CD-ROM. This is useful for standalone clients, because the diagnostics do not have to be installed on the local disk. Diagnostic support comes from a **SPOT** resource.

From Web-based System Manager

1. Select the NIM container.
2. In the contents area, select the standalone, diskless, or dataless machine you want to enable for diagnostics boot.
3. From the Selected menu, choose **Troubleshooting** —> **Enable Diagnostic Boot**.
4. Use the dialog to select a **SPOT** resource from which to boot.

From SMIT

Initiating the diag Operation from the Client

1. Enter the **smit nim_client_op** fast path.
2. Select the **diag** operation from the displayed list of operations.

Initiating the diag Operation from the Master

1. Enter the **smit nim_mac_op** fast path.
2. Select the machine object.
3. Select the **diag** operation from the list of operations.

From the Command Line

To perform the **diag** operation from the client, enter:

```
nimclient -o diag -a spot=SPOTName
```

To perform the **diag** operation from the master, enter:

```
nim -o diag -a spot=SPOTName
MachineObjectName
```

Verifying the diag Operation

After you have enabled the client to perform a diagnostic boot, you can verify the success of the operation by querying the client's *control state* (**Cstate**).

On the client, enter:

```
nimclient -l -l
ClientMachineObjectName
```

On the master, enter:

```
lsnim -l ClientMachineObjectName
```

If the operation is successful, output similar to the following is displayed:

```
Cstate = Diagnostic boot has been
enabled
```

For the client to boot the diagnostics, you need to reboot the client. If it is a diskless or a dataless client, you have already defined a network adapter as the default boot device (BOOTP request), so no additional action is required. For a standalone machine, the boot list for normal boot lists the hard disk as the primary boot device, so you must follow the "Booting a Machine Over the Network" on page 159 procedure .

Loading Diagnostics without the diag Operation

In addition to using the procedure in the previous section, diskless and dataless clients have another way of loading diagnostics from the network. You can boot a diskless or dataless client from the network the same way you do for normal use, but with the machine's key mode switch in the Service position. If the

client's key mode switch is in the Service position at the end of the boot process, hardware diagnostics from the server's **SPOT** are loaded. If a standalone client boots with the key mode switch in the Service position, the diagnostics (if installed) are loaded from the hard disk.

Booting in Maintenance Mode

If you need to perform maintenance on a standalone machine that is not part of the NIM environment, the system must be booted from a bootable tape or CD-ROM. This may require connecting an external device. If the machine is part of a NIM environment, you can enter maintenance mode directly by enabling the **maint_boot** operation for a NIM standalone machine.

From Web-based System Manager

1. Select the Machines container.
2. In the contents area, select a target standalone machine you want to enable for maintenance boot.
3. From the Selected menu, choose **Troubleshooting** —> **Enable Maintenance Boot**.
4. Use the dialog to select a **SPOT** resource from which to boot.

From SMIT

Initiating the maint_boot Operation from the Client

1. Enter the **smit nim_client_op** fast path.
2. Select the **maint_boot** operation.
3. Select the **SPOT** to be used for the operation.
4. Press Enter to enable the client for maintenance boot.

Initiating the maint_boot Operation from the Master

1. Enter the **smit nim_mac_op** fast path.
2. Select the client's machine object.
3. Select the **maint_boot** operation.
4. Select the **SPOT** to be used for the operation.
5. Press Enter to enable the client for maintenance boot.

From the Command Line

To issue the **maint_boot** operation from the client, enter:

```
nimclient -o maint_boot -a
spot=SPOTNAME
```

To issue the **maint_boot** operation from the master, enter:

```
nim -o maint_boot -a spot=SPOTNAME
CLIENT
```

To verify that the maintenance boot operation worked:

1. On the client, enter:

   ```
   nimclient -l -l ClientMachineObjectName
   ```

2. On the master, enter:

   ```
   lsnim -l  ClientMachineObjectName
   ```

If the operation was successful, the client's **Cstate** output will look similar to the following:

```
Cstate = maintenance boot has been
enabled
```

For the machine to boot into maintenance mode, follow the procedure for issuing the BOOTP request from the client. See “Booting a Machine Over the Network” on page 159 for more information about initiating a BOOTP request.

Using Maintenance Mode

After successfully booting and defining the console, the System Maintenance menu is displayed. The maintenance menu options and their descriptions are described below. Detailed information about maintenance mode can be found in the *AIX Version 4.3 Installation Guide*.

Access a Root Volume Group	This option allows you to activate the root volume group and start the maintenance shell with a full set of commands.
Copy a System Dump to Removable Media	This option allows you to copy a previous system dump to external media.
Access Advanced Maintenance Function	This option allows you to start a maintenance shell with a limited set of commands.

Chapter 7. Additional Topics

This chapter describes Network Installation Management (NIM) topics that are not part of the usual installation procedures.

The following topics are included:

NIM Master Management Tasks

The following tasks can be performed on the NIM master:

Deactivating the NIM Master and Removing the NIM Master Fileset

Once the NIM master fileset has been installed, the master activated, and the master object defined in the NIM database, this object, and hence the master fileset itself, cannot be removed. The master must be deactivated before the NIM master fileset can be removed.

To deactivate the master using Web-based System Manager, see "Unconfiguring the NIM Master" on page 59.

To use the command line to deactivate the master and remove the NIM master fileset, enter:

```
nim -o unconfig master
installp -u bos.sysmgt.nim.master
```

Increasing the Number of Hosts to Which NIM May NFS-Export a Resource

By default, when NIM exports a file or directory via NFS during resource allocation, it creates an entry in the **/etc/exports** file granting the target host both client mount access and root access for root users. As a result, when exporting to numerous clients, the limit on the length of a line in the exports file (32767 characters) may be exceeded, resulting in failure.

NIM provides an option to decrease the line length of an allocation entry in an NFS exports file by roughly half, effectively permitting files to be allocated to a greater number of hosts. This has the side effect of increasing the number of machines permitted in a NIM machine group. NIM achieves this by only granting root access to allocation target hosts. The client mount access list is not created which allows any machine to mount the resource, but still restricts root access to NIM clients only. NFS permits no more than 256 host names in a root exports file entry.

To enable this mode of operation, set the **restrict_nfs_exports** attribute to `no` on the master's NIM object. Use the **change** operation as follows:

```
nim -o change -a restrict_nfs_exports=no
master
```

To restore client mount access restrictions, set **restrict_nfs_exports** to `yes` with the **change** operation.

For information about how to export NIM resources globally, see "Exporting NIM Resources Globally" on page 69.

Controlling the Asynchronous Behavior of NIM Operations

Certain NIM operations are asynchronous, whereby the **nim** command executed on the master initiates the operation on the client, but does not wait for the operation to finish. This is because the NIM operation on the client is typically time-consuming. An example of an asynchronous operation is the **bos_inst** operation. Examples of synchronous operations are the **cust**, **maint**, and **lppchk** operations on a single machine target. However, these operations when applied to members of a machine group, are asynchronous. The **nim** command initiates these operations on each member of the group without waiting for the operation to finish.

If desired, the asynchronous behavior of **cust**, **maint**, and **lppchk** can be controlled by setting the **async** attribute on the command line. For example, to ensure that the execution of a customization script identified by the NIM resource `script1` is executed completely on a given member of the group `MacGrp1` before initiating execution of the script on the next member of the group, enter the following:

```
nim -o cust -a script=script1 -a async=no
MacGrp1
```

To force the master to not wait for the customization operation to finish when running the script on machine `Standalone1` that is not part of a machine group, enter:

```
nim -o cust -a script=script1 -a
async=yes Standalone1
```

Suppressing Output from NIM Operations

By default, progress messages are displayed by the **nim** command operating on machine groups to inform the user of how much processing remains. Similarly, the output from the installation and customization programs invoked by the **cust** and **maint** operations on **SPOTs** and machines is also displayed. This output can be suppressed by setting the **show_progress** attribute to `no` on the command line. For example, to tell NIM not to display output from the i**nstallp** command when updating the machine `Standalone1` with software from the **lpp_source** named `images1`, enter the following command:

```
nim -o cust -a show_progress=no -a
lpp_source=images1 \
-a fixes=update_all Standalone1
```

Reducing Space Requirements for NIM Resources

It is not unusual for resources such as the **SPOT** and **lpp_source** to take several hundred megabytes of storage space on a NIM server. By creating **/usr SPOTs** and defining CD-ROM file system directories as **lpp_sources**, space consumption can be reduced significantly on resource servers.

A **/usr SPOT** can be created from the **/usr** file system of the NIM master or any NIM client. The AIX system files for the Base Operating System are already installed, so only software for additional device support will be added to the system. The resulting system ultimately has more software installed on it than it needs to run, but far less disk space is used than otherwise would have been, had a **non-/usr SPOT** been created on the same system. For more information on creating **/usr SPOT** resources, see “SPOT (Shared Product Object Tree) Resource” on page 96 and “Defining /usr versus non-/usr SPOTs” on page 53.

A directory on the AIX product CD can be mounted and defined as an **lpp_source**, eliminating the need to copy installation images to the hard disk of a resource server. The defined **lpp_source** contains all the images available on the CD, but the CD must remain mounted at the server for the **lpp_source** to be usable in NIM operations. See “Defining an lpp_source on CD-ROM versus Disk” on page 55 for more information about using a CD-ROM file system as an **lpp_source**.

NIM Name Resolution

NIM relies on standard AIX library routines to perform name resolution. If a network environment uses multiple sources for name resolution, NIM will resolve host names by querying the sources in whatever order is specified for the system. For example, if a system is configured to resolve host names by first querying NIS, then BIND/DNS, then a local **/etc/hosts** file, NIM will also follow that order when resolving client host names.

Problems may result if the NIM master and the NIM clients use different orders when querying sources for name resolution. Problems may also arise if a name service is available to one machine but not to another, causing different name resolution sources to be used.

> **Note:** Mixing BIND/DNS, which is not case-sensitive, with NIS, which is case-sensitive, may result in problems.

It is possible to override the default system-wide order that AIX and NIM use when querying sources for host name resolution. This can be done by setting the **NSORDER** environment variable in the environment where NIM commands are being run. For example, to configure the environment to query NIS first, then BIND/DNS, then a local **/etc/hosts** file, type the following on the command line where NIM operations are being run:

```
export NSORDER=nis,bind,local
```

For more information on TCP/IP name resolution, refer to *AIX 5L Version 5.1 System Management Guide: Communications and Networks*.

Booting a FDDI Interface Over a Router

Boot over a router on a FDDI interface only if the router supports all-route broadcast. Booting over a router that does not support all-route broadcast on a FDDI interface may fail due to known limitations of these router types.

Default Paging Space During BOS Installation Via NIM

In AIX 4.3, default paging space is set by the BOS installation process when installing via NIM. The occurs, however, only if the following conditions are met:

- The method of installation is **overwrite**.
- Neither an **image_data** resource nor an **image.data** file on the diskette is specified for the installation.
- The source of the BOS image is neither a **mksysb** image nor a **SPOT**. A default paging space is set if the source of the BOS image is a **SPOT**, and:
  - the default **image.data** file contains more than one entry for paging. This file is located at:
    `(spot_location)/lpp/bosinst/image_template`
  - or if the LPs value for the single paging entry is set to the default value of **16**.

The default paging size is calculated from the smaller value of **optimal_ps** and **recommended_ps** where:

RAM = amount of memory on the target system measured in megabytes (MB).

optimal_ps = maximum between **RAM** and (0.2 size of rootvg)

IF CDE (COSE Desktop Environment) is installed, **recommmended_ps** =

- amount of **RAM** is less than 32MB, then **recommended_ps** = 3 * **RAM**
- amount of **RAM** is 32MB or more, then **recommended_ps** = **RAM** + 64MB

IF CDE (COSE Desktop Environment) is not installed, **recommmended_ps** =

- amount of **RAM** is less than 32MB, then **recommended_ps** = 2 * **RAM**
- amount of **RAM** is 32MB or more, then **recommended_ps** = **RAM** + 32MB

The default paging space set by this process is never greater than 512MB.

Migrating Diskless and Dataless Clients and NIM SPOTS

Migration to a new release of AIX is not currently supported for diskless and dataless clients. Also, migration of a **SPOT** that is not a converted **/usr** file system is not supported.

After migrating a machine that is a **SPOT** server to a new release of AIX, you must remove and redefine the **SPOT** in order to also bring it to the new AIX level.

To remove and redefine the **SPOT**, enter:

```
nim -o remove SPOT_name
nim -o define -t spot -a location=SPOTDirectory \
-a server=SPOTServer -a source=SPOTSource SPOTName
```

A **/usr SPOT** served by a client in the NIM environment can be reinstalled with a new level of AIX using the migration procedure, but the **SPOT** object must be removed and then redefined after the migration completes. Any diskless or dataless clients served by that **SPOT** must be reinitialized. To reinitialize diskless and dataless clients after migrating a **/usr SPOT** server, deallocate, then reallocate the root resources, and then perform the **dtls_init** or **dkls_init** operation accordingly.

To reinitialize diskless and dataless clients, enter:

```
nim -o reset -F ClientName
nim -o deallocate -a root=RootResourceName ClientName
nim -o allocate -a root=RootResourceName ClientName
nim -o dkls_init ClientName
```

Attention: Any customization that was done previously will be erased, because deallocating the root resource will delete all the files in the root directory.

Defining the NIM Environment Using the nimdef Command

The **nimdef** command assists administrators when defining complex NIM environments and adding large numbers of client machines.

The **nimdef** command solves a common usability problem when defining large NIM environments.

Regardless of how well a NIM environment is understood, it can be a very time-consuming process to execute all the commands necessary to define it. If NIM could process a simple definition file for configuration of the NIM environment, a great deal of time could be saved that would otherwise be spent defining each network and machine manually.

The **nimdef** command reads a definition file for input. The definition file is in a structured stanza format. Each stanza describes a machine that will be added to the NIM environment. Included in the stanza is information about the machine's network adapter and routing configuration. Based on the supplied information, the **nimdef** command can determine the remaining information needed to define both networks and machines in the NIM environment.

For more information, see the **nimdef** Command. See "Appendix C. Sample Files" on page 165 for a sample definition file for the **nimdef** command.

Name Requirements for NIM Object Definitions

The name that you give a NIM object will be used in all future operations involving that object. This name must be unique among NIM objects, and it must adhere to certain restrictions:

- It must have between 1 and 39 characters.
- Valid NIM name characters include the uppercase and lowercase letters of the alphabet, the numbers 0-9, and the underscore character.
- Invalid NIM name characters include the dot character, all shell metacharacters, all file system metacharacters, and all regular expression metacharacters.

Interacting with the Dynamic Host Configuration Protocol

You should select your NIM master to be the same system as the Dynamic Host Configuration Protocol (DHCP) server when using NIM in an environment that uses DHCP. You should also use host names whenever possible when defining NIM machine objects.

Refer to DHCP and Network Installation Management (NIM) Interactions and Suggestions in *AIX 5L Version 5.1 System Management Guide: Communications and Networks* for more information.

Creating File Resources in the root Directory

Due to a limitation in NFS, file resources, such as **bosinst_data** and **script** resources cannot be created in the root directory ("**/**") of a resource server.

Restricting NIM Client Resource Allocation

NIM provides client machines with the capability of allocating and using any resource in the NIM environment. In some tightly controlled NIM environments, administrators may not want clients to be able to access all resources at all times. To control client-resource allocation, a NIM administrator can use the **client_alloc** attribute. The restrictions placed by the **client_alloc** attribute will prevent clients from allocating and using resources, but the NIM master will continue to have the full capability of performing operations on clients.

Note: This task is not currently supported by Web-based System Manager.

From SMIT

NIM client-allocation restrictions can be changed from the SMIT interface by typing the SMIT fast path:

```
smit nim_control_alloc
```

From the Command Line

To restrict all clients from being able to use any resources, set the attribute **client_alloc=no** on the NIM master:

```
nim -o change -a client_alloc=no master
```

To restrict a particular client from being able to use any resources, set the attribute **client_alloc=no** on the client:

```
nim -o change -a client_alloc=no clientname
```

To restrict all clients from being able to use a particular resource, set the attribute **client_alloc=no** on the resource:

```
nim -o change -a client_alloc=no resourcename
```

To lift the restrictions on client-resource allocation, remove the **client_alloc** attribute by setting it to ″yes″ for the applicable object:

```
nim -o change -a client_alloc=yes master
nim -o change -a client_alloc=yes clientname
nim -o change -a client_alloc=yes resourcename
```

Preventing Machines from Adding Themselves as Clients

Machines may add themselves as clients in NIM environments by using the **niminit** command and specifying the hostname of a NIM master. In some environments, administrators may want total control over which machines are added as clients of their masters. To prevent clients from adding themselves to a NIM environment, an administrator can use the **client_reg** attribute.

Note: This task is not currently supported by Web-based System Manager.

From SMIT

The option to allow clients to add themselves to a NIM environment can be changed from the SMIT interface by typing the SMIT fast path:

```
smit nim_client_reg
```

From the Command Line

To prevent machines from adding themselves as clients in a NIM environment, set the attribute **client_reg=no** on the NIM master:

```
nim -o change -a client_reg=no master
```

To allow machines to add themselves as clients of a NIM master, remove the **client_reg** attribute by setting it to ″yes″ on the master:

```
nim -o change -a client_reg=yes master
```

Disabling Client CPU ID Validation

The CPU ID of a NIM client is stored in the NIM database so that the master can perform verification that NIM client commands are coming from the machines that were originally registered as clients. There are two situations when a NIM administrator would not want this CPU ID validation to be performed:

- When the hardware of a client machine is changed, giving the client a new CPU ID.
- When a single client definition is used to install different machines, as on a preinstall assembly line.

From Web-based System Manager

To enable or disable NIM client CPU ID validation from the NIM application:

1. From the NIM menu, select **Advanced Configuration** > **Control Client CPU ID Validation**.
2. Use the dialog to complete the task.

From SMIT

Manage the client CPU ID validation from the SMIT interface by typing the SMIT fast path:

```
smit nim_cpuid_validate
```

From the Command Line

Client CPU ID validation can be managed on the NIM master by using the **validate_cpuid** attribute.

To disable client CPU ID validation, set the attribute **validate_cpuid=no** on the NIM master:

```
nim -o change -a validate_cpuid=no master
```

To perform client CPU ID validation, remove the **validate_cpuid** attribute from the master by setting it to ″yes″:

```
nim -o change -a validate_cpuid=yes master
```

> **Attention:** The value of the **validate_cpuid** attribute should not be changed while operations are being performed on NIM clients because it could potentially disrupt client communications for active machines.

Exporting NIM Resources Globally

When resources are allocated for use during NIM operations, they are NFS-exported to the client machines where the operations will be performed. If operations are performed simultaneously on many different clients, the **/etc/exports** and **/etc/xtab** files may become very large on the resource servers. This may cause size limits to be hit in the files, and it may also negatively affect NIM performance as the files are locked and modified for each resource allocation/deallocation.

In environments where administrators are not concerned about who has access to the NIM resources, they may set an option to globally export the resources and thereby eliminate the repeated updates to the **/etc/exports** and **/etc/xtab** files. The only resources that may not be globally exported are those that are used exclusively by diskless and dataless clients. The global export of a NIM resource will make it readable by any machine in the network, not just those in the NIM environment. The resource will be globally exported as long as it is allocated to any client. When the resource is deallocated from all clients, it is unexported. Global exporting is only supported for resource servers that have the NIM client fileset installed at level 4.3.0 or higher.

From Web-based System Manager

To enable or disable global export of NIM resources from the NIM application:

1. From the NIM menu, select **Advanced Configuration** > **Export NIM Resources Globally**.
2. Use the dialog to complete the task.

From SMIT

Manage global exporting of NIM resources from the SMIT interface by typing the SMIT fast path:

```
smit nim_global_export
```

From the Command Line

Global exporting of NIM resources for use by clients can be managed with the **global_export** attribute.

To enable global exporting of NIM resources, set the attribute **global_export=yes** on the NIM master:

```
nim -o change -a global_export=yes master
```

To disable global exporting of NIM resources, remove the **global_export** attribute from the master by setting it to ″no″:

```
nim -o change -a global_export=no master
```

The enablement and disablement of global exports should not be changed when there are resources allocated to clients. This could lead to situations where resources are exported with incorrect permissions. All NIM operations should be completed and resources deallocated before any attempts are made to change the **global_export** value. The nim command to change the **global_export** value will fail if resources are currently allocated to clients.

Creating Network Boot Images to Support Only the Defined Clients and Networks

When a SPOT resource is created, network boot images are created in the **/tftpboot** directory to support certain NIM operations. Prior to AIX 4.3, the default behavior of NIM was to create a network boot image for every type of machine and network for which support was available in the SPOT. This resulted in the creation of many unneeded boot images that used a large amount of disk space and slowed down the SPOT creation and check operations.

In AIX 4.3, NIM's default behavior has been changed to only create network boot images to support clients and networks that are defined. If a new client is defined and there is no network boot image already created for it in the environment, then the boot image will not be created until either the SPOT is allocated to the client or a check operation is performed on the SPOT to rebuild the boot images.

When clients are removed from the NIM environment, boot images are not automatically removed. To remove boot images that are no longer necessary for a NIM environment, the list of required machine-network combinations in the environment must be rebuilt. The boot images must then be rebuilt for each SPOT.

From Web-based System Manager

To limit or enable boot image creation according to whether the interface is defined from the Web-based System Manager application:

1. From the NIM menu, select **Advanced Configuration** > **Control Network Boot Image Creation**.
2. Use the dialog to complete the task.

From SMIT

Manage the creation of boot images from the SMIT interface by typing the SMIT fast path:

```
smit nim_control_boot
```

From the Command Line

To rebuild the list of machine types and networks that must be supported by network boot images in the NIM environment, perform a **change** operation on the NIM master with the **if_discover=yes** attribute:

```
nim -o change -a if_discover=yes master
```

To rebuild network boot images from a SPOT, perform a **check** operation on the SPOT with the **force** option:

```
nim -Fo check spot name
```

If an administrator prefers to have NIM always create all possible boot images from the SPOT resources, the **if_prebuild=yes** attribute can be specified on the master:

```
nim -o change -a if_prebuild=yes master
```

To return NIM to the behavior of creating only the boot images that are required for the environment, remove the **if_prebuild** attribute from the master by setting it to ″no″:

```
nim -o change -a if_prebuild=no master
```

Updating a Spot With New Device Support for a New Level of AIX

A NIM SPOT may be updated from one level of AIX to another using the **update_all** option of NIM's **cust** operation. This process will update all currently SPOT with the lates level of code on the installation media. However, this process will not automatically install new software packages or device drivers from the installation media.

Machines in the NIM environment that are being upgraded to a new level of AIX will require that new applicable device support be updated for any existing NIM SPOTs intended to support network boot and installation. This must be done after the SPOT is updated to the new level of AIX.

The new device support can be installed in the SPOT using NIM's **cust**operation, specifying the desired device specific filesets in an **installp_bundle** resource or by using the **filesets** attribute. Alternatively, a fileset name of **devices** can be specified as the value of the filesets attribute to inatall all devices on the installation media. See “cust” on page 109 for further details on the cust operation.

Tune Client Request Processing

For large installation environments, NIM can be scaled to support anywhere from 20 to 150 client requests simultaneously. This is done by enabling the multi-threaded option on the nimesis daemon. This option provides better handling of the volume of client information change requests and client state changes. Without the use of this option the NIM master machine can become over-loaded by activity on the NIM database and the number of active processes, resulting in failures during the installation of a large number of client machines simultaneously.

The multi-threaded nimesis daemon will serialize and buffer nimclient requests to protect the NIM master machine from process over-load, without causing significant performance degradation. The user must understand that many of the client information changes will not be reflected in the NIM database. The most recent information change for any client, however, are eventually processed. Debugging failed or hung clients will not be adversely affected.

The number of threads assigned to this daemon determines how many simultaneous NIM client requests can be handled in the NIM environment. Since most of the nimclient requests are processed rapidly, it is not necessary to have one thread for every client installing. The number of threads needed to support the activities in a NIM environment is dependent upon several items. The following should be considered when determining the number of threads:

- number of clients that will be operated on at the same time
- processing capacity of the NIM master machine
- what type of operations are planned

In general, one thread should support two to four clients that are installing BOS at the same time. For example, when installing 150 machines, 50 to 75 threads should be sufficient. The number of threads is highly dependent on the processing power of the NIM master machine, and slower master machines may require more threads.

For smaller NIM environments, enabling the multi-threaded daemon can tie up system resources on the master that will not be utilized. For example, when installing 50 machines simultaneously, 20 to 25 threads or even the single-threaded daemon (this option disabled) would suffice.

> **Note:** This option alone will not allow more machines to be installed simultaneously. This option should be used in conjunction with global export of NIM resources, distribution of NIM resources throughout the NIM environment, and a network environment capable of handling a large volume of throughput.

From SMIT

To tune client request processing from the SMIT interface, type the SMIT fast path:

```
smit nim_tune_nimesis
```

From Web-based System Manager

To tune client request processing from the NIM application:

1. From the NIM menu, select **Advanced Configuration—>Tune Client Request Processing.**
2. Use the dialog to complete the task.

From the Command Line

The **max_nimesis_threads** attribute can be used to tune client request processing. To enable the multi-threaded nimesis daemon, set a value to the attribute **max_nimesis_threads** on the NIM master using the following command:

```
nim -o change -a max_nimesis_threads=value master
```

> **NOTE:** The range for the *value* attribute above is 20 to 150.

To disable the multi-threaded nimesis daemon, set a null value to the attribute **max_nimesis_threads** on the NIM master:

```
nim -o change -a max_nimesis_threads="" master
```

Chapter 8. Network Installation Management Concepts

This chapter discusses the concepts required to understand the operation of Network Installation Management (NIM). NIM uses the AIX 4.2 (and later) installation process to install systems by using the network. To use all the available features in NIM, you should understand various components of AIX installation. The details discussed in this chapter focus on command line operations, but the information is applicable to the other NIM interfaces, as well. Use this chapter as reference material to supplement the online help available in the other interfaces.

This section contains the following topics:

NIM Machines

There are currently three types of machines that can be managed in the NIM environment. These are *standalone*, *diskless*, and *dataless* clients. This section describes the differences between the machines, the attributes required to define the machines, and the operations that can be performed on them. 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 other machines participating in the NIM environment are clients to the master, including machines that may also serve resources.

NIM Operations on Client Machines

There are unique operations to initialize the different client configurations. NIM uses this fact to check that the operation is a valid operation for a specific client configuration. The following table shows the operations that can be performed on the different client configuration types.

NIM Operation	Machine Configuration		
	Standalone	Diskless	Dataless
bos_inst	x		
dkls_init		x	
dtls_init			x
diag	x	x	x
cust	x		
fix_query	x		
lppchk	x		
maint	x		
maint_boot	x		
reset	x	x	x
check	x	x	x
showlog	x	x	x
reboot	x	x	x

For more information about NIM operations, see “NIM Operations” on page 101.

Defining NIM Clients

Standalone, diskless, and dataless clients are defined in the NIM environment using the NIM **define** operation. The command line syntax is as follows:

```
nim -o define -t MachineType -a
Attribute=Value ... MachineName
```

where the following attributes are required:

-t *MachineType*
: Specifies the type of machine being defined. Valid values are **standalone**, **diskless**, and **dataless**.

-a if=*Value ...*
: Stores network interface information for a NIM client, and requires a sequence number when specified. The value for this attribute consists of three required values and a fourth, optional value:

  Value 1
  : Specifies the name of the NIM network to which this interface connects. If the name of the NIM network is unknown, then the **find_net** keyword can be used to match the client's IP address to a defined NIM network. If the **find_net** keyword is used, but NIM does not find a matching network, the optional **net_definition** attribute should be used to define the network, as well.

  Value 2
  : Specifies the host name associated with this interface.

  Value 3
  : Specifies the network adapter hardware address of this interface. A value of **0** can be specified unless broadcasting is used for network boot of the client.

  Value 4
  : Specifies the logical device name of the network adapter used for this interface. If this value is not specified, NIM uses a default based on the type of network interface defined. This field is required when the client is defined on a heterogeneous network.

  This attribute requires a sequence number for NIM to distinguish between multiple network interfaces. Because machines can be multihomed, NIM allows more than one **if** attribute per machine.

The following attributes are optional:

-a ring_speed=*Value*
: Specifies the ring speed of the client's token-ring adapter. This value is required if the client's NIM network is token-ring. This attribute requires a sequence number for NIM to distinguish between ring speeds for multiple interfaces on the machine.

-a cable_type=*Value*
: Specifies the cable type of the client's ethernet adapter. This value is required if the client's NIM network is ethernet. This attribute requires a sequence number for NIM to distinguish between cable types for multiple interfaces on the machine.

-a platform=*Value*
: Specifies the platform of the machine being defined. The default value is **platform=rs6k**. Other examples are **rspc** and **chrp**. Run the **bootinfo -p** command on a running machine to determine its platform.

-a netboot_kernel=*Value*
: Specifies the kernel type of the client. Valid values are **up** for uniprocessor machines and **mp** for multiprocessor machines. The default value is **netboot_kernel=up**.

-a iplrom_emu=*Value*
Specifies the device that contains the IPL ROM emulation software. IPL ROM emulation is required for machines that do not have bootp-enabled IPL ROM.

-a net_definition=*Value ...*
Defines a NIM network to be associated with the client being defined. The value for this attribute consists of two required values and three optional values:

Value 1 = NetworkType **(required)**
Specified values are **tok**, **ent**, **fddi**, and **generic**.

Value 2 = SubnetMask **(required)**
Specifies the dotted decimal mask for the network.

Value 3 = ClientGateway **(optional)**
Specifies the IP address or host name of the default gateway used by the machine being defined to communicate with the NIM master.

Value 4 = MasterGateway **(optional)**
Specifies the IP address or host name of the default gateway used by the NIM master to communicate with clients on other subnets.

Value 5 = NetworkName **(optional)**
Specifies a name to be given to the NIM definition created for the network. (Otherwise, a unique default value is assigned.)

When specifying the **net_definition** attribute to create or change a machine definition, the **find_net** keyword must be specified as the first component of the **if** attribute for the machine. The **net_definition** attribute may also be specified when defining additional NIM interfaces (**if** attributes) for machine definitions.

-a cpuid=*Value*
Specifies the CPU ID of the machine being defined. This attribute can be used for client verification during NIM operations. To display the CPU ID on a running machine, use the **uname -m** command. This field is optional and will be automatically set the first time a client communicates with the NIM master.

-a master_port=*Value*
Specifies the port number used by the NIM master for socket communication with the clients. The default master port number is **1058**.

-a registration_port=*Value*
Specifies the port number used by clients to register themselves with the NIM master. The default registration port number is **1059**.

-a group=*Value*
Specifies a machine group to which the client should be added. The group will be defined if it does not exist.

-a comments=*Value*
Provides comments about the client being defined.

-a verbose=*Value*
Displays information for debugging. Use **verbose=5** to show maximum detail.

Standalone Clients

Standalone NIM clients are clients with the capability of booting and running from local resources. Standalone clients mount all file systems from local disks and have a local boot image. Standalone clients are not dependent upon network servers for operation.

Network Booting a Standalone Client

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

order for NIM to perform a BOS installation (**bos_inst**) of the client or to boot into maintenance mode (**maint_boot**) and diagnostics (**diag**). For instructions on booting a client over the network, see “Booting a Machine Over the Network” on page 159.

Managing Software on Standalone Clients

The AIX Base Operating System can be installed directly on standalone clients using the NIM **bos_inst** operation. Additional software and updates can be installed and managed on standalone clients using the NIM **cust** and **maint** operations. See “NIM Operations” on page 101 for more information about these and other operations.

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. As their names imply, diskless clients have no hard disk, and dataless clients have disks that are unable to hold all the data that may be required for operation. Diskless machines must mount paging space and all file systems from remote servers. Dataless machines can only use a local disk for paging space and the **/tmp** and **/home** file systems. Neither diskless nor dataless clients have a local boot image, and they must boot from servers on the network.

Diskless/dataless clients and resources are not supported for platforms of type Itanium-based.

The reasons for defining a machine as diskless or dataless are:

- **Cost savings**

  No hard disk is required for diskless clients. Only a small hard disk is needed for dataless clients.
- **Manage software configurations on machines**

  On diskless and dataless clients, the file system containing the Base Operating System is mounted from a server. All client systems that mount the same file system for BOS run from identical software.
- **Manage storage of user data**

  User data for diskless and dataless clients are stored on remote servers. A system administrator can manage storage allocation and data backups for the client machines by managing the data on the server, rather than on each machine separately.

Required and Optional Resources for Diskless and Dataless Clients

The file systems that are mounted by the diskless and dataless client machines are treated as resources in the NIM environment. Like other resources, they exist on a server in the NIM environment, and they are NFS-exported to the clients that use them.

The following resources are managed by NIM to support diskless and dataless clients:

boot
: Defined as a network boot image for NIM clients. The **boot** resource is managed automatically by NIM and is never explicitly allocated or deallocated by users.

SPOT
: Defined as a directory structure that contains the AIX run-time files common to all machines. These files are referred to as the **usr** parts of the fileset. The **SPOT** resource is mounted as the **/usr** file system on diskless and dataless clients.

  Contains the **root** parts of filesets. The **root** part of a fileset is the set of files that may be used to configure the software for a particular machine. These **root** files are stored in special directories in the **SPOT**, and they are used to populate the root directories of diskless and dataless clients during initialization.

  The network boot images used to boot clients are constructed from software installed in the **SPOT**.

  A **SPOT** resource is required for both diskless and dataless clients.

root Defined as a parent directory for client ″**/**″ (**root**) directories. The client root directory in the **root** resource is mounted as the ″**/**″ (**root**) file system on the client.

When the resources for a client are initialized, the client **root** directory is populated with configuration files. These configuration files are copied from the **SPOT** resource that has been allocated to the same machine.

A **root** resource is required for both diskless and dataless clients.

dump Defined as a parent directory for client dump files. The client dump file in the **dump** resource is mounted as the dump device for the client.

A **dump** resource is required for both diskless and dataless clients.

paging Defined as a parent directory for client paging files. The client paging file in the **paging** resource is mounted as the paging device for the client.

A **paging** resource is required for diskless clients and optional for dataless clients.

home Defined as a parent directory for client **/home** directories. The client directory in the **home** resource is mounted as the **/home** file system on the client.

A **home** resource is optional for both diskless and dataless clients.

shared_home Defined as a **/home** directory shared by clients. All clients that use a **shared_home** resource will mount the same directory as the **/home** file system.

A **shared_home** resource is optional for both diskless and dataless clients.

tmp Defined as a parent directory for client **/tmp** directories. The client directory in the **tmp** resource is mounted as the **/tmp** file system on the client.

A **tmp** resource is optional for both diskless and dataless clients.

resolv_conf Contains nameserver IP addresses and a network domain name.

Unlike the other resources used by diskless/dataless clients, the **resolv_conf** resource does not remain mounted by the client. Instead, it is copied to the **/etc/resolv.conf** file in the client's root directory.

A **resolv_conf** resource is optional for both diskless and dataless clients.

Initialization of Diskless and Dataless Clients

Diskless and dataless clients are not installed in the same way as standalone machines. Instead, they are initialized. Initialization of diskless and dataless clients involves several phases of operation:

Resource Allocation The resources required to support a diskless/dataless client must be allocated to the client before or during the initialization operation.

If the resource is a parent directory of client directories, the allocation will create an empty subdirectory for the client. The client subdirectory is then NFS-exported to the client. The client subdirectories are not populated until the initialization is actually performed.

Client Initialization

The **dkls_init** and **dtls_init** operations are used in NIM to initialize the resources for client use.

Among the operations performed during client initialization are the following:

- The boot image is made available to the client for performing a network boot.
- The root files, which are used for machine-specific customization, are copied into the client's subdirectory in the **root** resource. The files that are copied into the client root directories come from the **SPOT** resource that has been allocated to the client.
- The **/tftpboot/***Client***.info** file is created on the boot server (which is the **SPOT** server). This file contains information that will be needed by the client during the start-up configuration processing to successfully configure as a diskless or dataless client.

  The following are some of the variables defined in the *Client***.info** file:

  ```
  export NIM_CONFIGURATION=diskless
  export RC_CONFIG=rc.dd_boot
  export ROOT=Host:Client_Root_Directory
  export DUMP=Host:Client_Dump_Directory
  export SPOT=Host:SPOT_Location
  ```

  The paging location is set in the client's root directory in the **/etc/swapspaces** file.

Network Boot of the Client

The client machine is booted over the network using standard **bootp** procedures for the machine type. The client obtains the boot image and begins running a mini-kernel in a file system in RAM.

The client tftp's the *Client***.info** file from the **/tftpboot** directory on the **SPOT** server. The information in the *Client***.info** file is used to properly configure the client as a diskless or dataless machine.

The remote file systems are mounted from the resource servers.

If the client is a dataless client, and no **paging**, **tmp**, **home**, or **shared_home** resource is allocated, then the client will create the missing file system on the local hard disk.

Managing Software on Diskless and Dataless Clients

The **/usr** and **root** file systems of diskless and dataless clients are resources that have been mounted from a server. Therefore, in order to install or deinstall software on a diskless/dataless client, the processing must actually occur on the resources that the clients use.

The **SPOT** contains the directory structure for an installed **/usr** file system. It also contains subdirectories for the ″root″ parts of installed filesets. Since the **SPOT** contains both **usr** and **root** files, software maintenance must be performed on the **SPOT** in order to update the software that is running on the clients. Such actions must be performed using the NIM **cust** and **maint** operations. See "NIM Operations" on page 101 for more information about the **cust** and **maint** operations.

If the **SPOT** is currently allocated for client use, NIM will prevent software customization operations from being performed on it. This is to safeguard the **SPOT** from changes that may adversely affect running client machines. However, this restriction can be overridden by specifying the **force** option when performing the operation.

When NIM is used to install software in a **SPOT**, the following operations are performed to manage the software for diskless and dataless clients:

1. The **/usr** files are installed in the **SPOT**. These files are automatically seen by all the clients that mount the **SPOT** as their **/usr** file systems.
2. The root files are installed in special subdirectories in the **SPOT**.
3. After all the filesets have been installed in the **SPOT**, the **root** files are copied to the **root** directories of any diskless or dataless clients that have been initialized with the **SPOT**.

When NIM is used to deinstall software in a **SPOT**, the following operations are performed to manage the software for diskless and dataless clients:

1. The **/usr** files are removed from the **SPOT**. This automatically ″removes″ the files from the client systems as well.
2. The **root** files of the software are removed from the client **root** directories.

NIM also provides a **sync_roots** operation to perform consistency verification and correction to ensure the client **root** directories match the **root** parts stored in the **SPOT**.

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 be able to verify that client machines can access all the resources required to support operations. To avoid the overhead of repeatedly specifying network information for each individual client, NIM networks are used to represent the networks in a NIM environment. When NIM clients are defined, the associated network for the client must be specified. During NIM operations, the NIM master is able to use information from the client's network definition when necessary.

When the NIM master is configured, the network associated with the master is automatically defined in the NIM environment. It is only necessary to define additional NIM networks if clients reside on other local area networks or subnets. The procedures described in this guide and reference are designed to automatically define NIM networks, if necessary, when clients are added. However, this section is included to describe NIM networks in detail in case manual definition of networks and routes is required.

Supported NIM Network Types

The currently supported network types are:

- Ethernet
- Standard Ethernet
- IEEE 802.3 Ethernet
- Token-Ring
- FDDI
- ATM
- Generic

Network boot support is provided for Ethernet, Token-Ring, and FDDI. Unlike other network adapters, ATM adapters cannot be used to boot a machine. This means that installing a machine over an ATM network requires special processing. See “Installing to Clients on ATM Networks” on page 21. The Generic network type is used to represent all other network types where network boot support is not available. For clients on Generic networks, NIM operations that require a network boot, such as **bos_inst** and **diag**, are not supported. However, nonbooting operations, such as **cust** and **maint**, are allowed. Diskless and dataless clients cannot be associated with Generic networks, since they inherently rely on network boot capability.

Defining NIM Networks

Networks are defined in the NIM environment using the NIM **define** operation. The command line syntax is as follows:

nim -o define -t NetworkType -a Attribute=Value ... MachineName

where the following attributes are required:

-a net_addr=*Value*
: Specifies the IP address of the network being defined. If the network address is not known, see “Determining a Network’s IP Address” on page 81.

-a snm=*Value*
: Specifies the subnet mask for the network.

-t *NetworkType*
: Specifies the type of network being defined. Valid values are **atm**, **tok**, **ent**, **fddi**, and **generic**.

The following attributes are optional:

-a comments=*Value*
: Provides comments about this network.

-a ieee_ent=*Value*
: Specifies IEEE 802.3 ethernet configuration. This is only valid for networks that are defined with the **ent** type or those that have an **other_net_type** attribute set to **ent**.

-a other_net_type=*Value*
: Specifies another network type that applies to this logical network. Each NIM network is used to represent one logical network that exists in the NIM environment. When the network is defined, the type of network interface used in the network must be supplied. Usually, a network is composed of only one type. However, a bridge can be used to connect different network types together to form one logical network. In that situation, NIM needs to know what the other network interface types are, and this attribute is used to specify that information. For more information on how to use the **other_net_type** attribute, see “Defining a Heterogeneous Network” on page 82.

-a routing=*Value* ...
: Stores NIM routing information for a network. This attribute requires a sequence number when specified. When a new NIM route is specified, the **routing** attribute consists of three values:

  Value 1
  : Specifies the NIM name of the destination network for this route.

  Value 2
  : Specifies the host name of the gateway to use in order to communicate with the destination network.

  Value 3
  : Specifies the host name of the gateway used by the destination network to get back to this network.

  This attribute can be used to add a default route or static route. To add a default route, specify **default** for *Value 1*. Then, specify the default gateway for the network in *Value 2*. Leave *Value 3* blank.

  For more information on adding and changing routes, see “NIM Routes” on page 81, “Establishing a Default NIM Route Between Networks” on page 56, and “Establishing a Static NIM Route Between Networks” on page 57.

-a verbose=*Value*
: Displays information for debugging. Use **verbose=5** to show maximum detail.

It is also possible to define NIM networks automatically when client machines are defined. To do this, use the **find_net** and **net_definition** attributes when defining the client. For more information, see “NIM Machines” on page 73.

Determining a Network’s IP Address

NIM determines a network’s IP address by performing a bitwise ″AND″ on the binary representations of the network’s subnet mask and the address of any machine’s IP address on the same network. For example:

```
    subnet mask = 255.255.254.0
 client address = 129.35.58.207

In binary:

    subnet mask = 11111111.11111111.11111110.00000000
 client address = 10000001.00100011.00111010.11001111
network address = 10000001.00100011.00111010.00000000

In decimal:

network address = 129.35.58.0
```

NIM Routes

Routing information is used internally by NIM to ensure that a client on one network can communicate with a server on another network. It defines the gateway to use to go from one network to the other network.

NIM provides the ability to define default or static routes. Default NIM routes provide the following advantages over static routes:

- They more closely model the network configuration of common network environments.
- They also permit resources that are distributed throughout a NIM environment to be more easily accessed by any client in the NIM environment.

Static NIM routes are supported for backward compatibility with NIM environments defined on machines running AIX 4.1 (and later).

To determine the gateway used by machines on a given network, run **netstat -rn** on a running machine on the network to see if a default gateway is listed. You can also issue **traceroute** *Host_Name* from a running machine on the network in question, where *Host_Name* is the name of the master’s primary network interface if determining the gateway for a client, or the name of a target client if determining the gateway used by the master. The first gateway listed is the gateway used by machines on the specified network.

Note that NIM routes are not required if the only networks defined in a NIM environment are associated with interfaces (**if** attributes) defined on the NIM master and if all resources will be defined on the master. If resources are served by machines other than the master to clients that do not reside on the same network as the server, NIM routes are required between those networks even if all networks are attached to interfaces belonging to the master. In this case, the master must act as a gateway (with IP-forwarding switched on), and the host name of the interface on the master should be used as a gateway.

Networks with default routes may be created automatically when NIM machines are being defined.

It may be observed that communications between networks go through several gateways. However, it is important to remember that when defining NIM routes for networks, the only gateways of interest are the first ones used by the networks to reach their destinations. Intermediate gateways between the originating and destination networks are irrelevant for NIM routing purposes.

For more information on adding and changing routes, see “Establishing a Default NIM Route Between Networks” on page 56 and “Establishing a Static NIM Route Between Networks” on page 57.

Defining a Heterogeneous Network

This section describes a feature of NIM that enables it to model networks consisting of different data link protocol segments. These kinds of networks use bridges to connect two segments that have different data link protocols. A network consisting of a Token-Ring and an Ethernet segment can be connected to form a single logical network, as shown in the following figure.

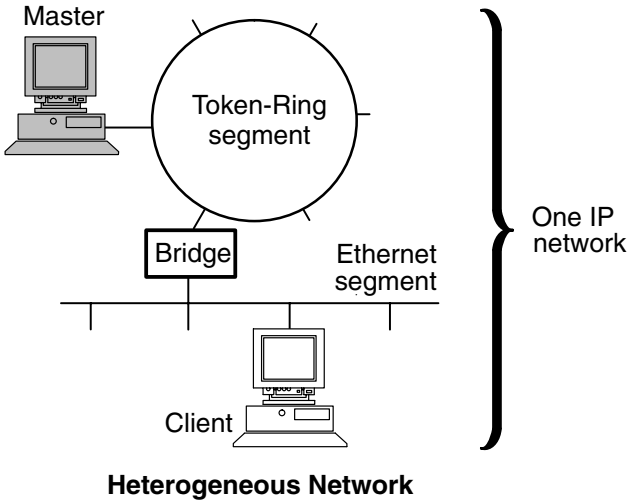

Figure 1. Heterogeneous Network. This illustration shows a single IP network in which the master server uses its token-ring connection and a bridge to communicate with its client on an Ethernet segment.

Because a single NIM network object is used to represent one network, the **other_net_type** attribute is reserved for a different type of interface that can exist in a network. The **other_net_type** attribute can be added to the definition of a network object. When present in a network definition, the **other_net_type** attribute tells NIM that this logical network uses a bridge to connect the other network type to the network type that was specified when the object was defined.

When you define a machine object to be connected to a network object, NIM checks to see if the network has any **other_net_type** attributes. If so, NIM requires that the fourth field, which is usually optional, in the **if** attribute, be specified. This field specifies the logical name of the client's network adapter. The following example defines a network object that has a bridge joining a Token-Ring and an Ethernet segment:

```
nim -o define -t tok -a
net_addr=129.35.129.0 \
    -a snm=255.255.240.0 -a other_net_type1=ent b905net

lsnim -l b905net

class           =  network
type            =  tok
net_addr        =  129.35.128.0
snm             =  255.255.240.0
other_net_type1 =  ent
Nstate          =  ready for use
prev_state      =  information is missing from this object's def>
```

Note that **other_net_type** requires a sequence number. This is because a network could be composed of all three types of interfaces linked by bridges.

When you define a client's interface that is physically connected to an Ethernet segment joined with a Token-Ring network using a bridge (with master being on the Token-Ring side), you must supply the fourth field:

```
nim -o define -t standalone -a
if1='find_net mymac 08005ac9430c \
ent' -a cable_type1=bnc mymac
```

Adding Another Network Type to a NIM Network

From Web-based System Manager

1. In the NIM Network container, double-click on a network. The General page of the properties notebook displays.
2. Use the General page to add a network type to the network.

From SMIT

1. To add another network type, enter the **smit nim_chnet** fast path.
2. Select the network to change.
3. Specify the additional network type to be supported.

From the Command Line

To define a NIM network, enter:

```
nim -o change -a
other_net_typeSequenceNumber=NetworkType \
NetworkName
```

For example, to change a Token-Ring network called `network1` to also support Ethernet and FDDI, enter:

```
nim -o change -a other_net_type1=ent -a
other_net_type2=fddi \
network1
```

NIM Resources

A large number of resources (files and directories) are needed to support NIM software installation and maintenance operations. Resources can be added through the Web-based System Manager **New Resources** wizard. The NIM resources are as follows:

boot	**image_data**	**resolv_conf**
bosinst_data	**installp_bundle**	**root**
dump>	**lpp_source**	**script** >
exclude_files	**mksysb**	**shared_home**
fix_bundle	**nim_script**	**SPOT**
home	**paging**	**tmp**

If you want to obtain detailed information about any resource, enter the following from the NIM master:

```
lsnim -Pa ResourceType
```

The following sections describe detailed information about each NIM resource. The Web-based System Manager and SMIT interfaces are designed to hide much of the detail required for the command line interface. Therefore, this section only documents the resource task procedures for the command line. The

following information applies to the other interfaces as well, but discussion of those interfaces is deferred to the online contextual help available for those applications.

boot Resource

The **boot** resource is an internally-managed NIM resource used to indicate that a boot image has been allocated to a client. The **boot** resource is automatically allocated to clients to support NIM operations requiring a network boot. The **boot** resource will be automatically deallocated when the operation completes.

bosinst_data Resource

A **bosinst_data** resource represents a file that contains information for the BOS install program. Normally, the BOS installation program looks for this information in the **/bosinst.data** file in the BOS install image. If this file does not exist or if it does not contain all the information that the BOS installation program requires, the program prompts for information by using a console that is local to the target. Information must then be specified manually for the BOS installation to proceed. With a **bosinst_data** resource, the data can be specified in a NIM resource prior to the installation to prevent the need for prompting at the console.

A sample **bosinst.data** file (*SPOT_Offset*/**usr/lpp/bosinst/bosinst.template**) is located on the **SPOT** resource server. Also, see “Appendix C. Sample Files” on page 165 for a sample **bosinst_data** file.

For instructions on how to create and use a **bosinst_data** file, see “Performing a Non-Prompted BOS Installation” on page 20.

Defining a bosinst_data Resource

The command line syntax for defining a **bosinst_data** resource is:

```
nim -o define -t bosinst_data -a Attribute=Value ... bosinst_dataName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the **bosinst_data** resource file.
-a server=*Value*	Specifies the name of the machine where the **bosinst_data** resource file resides.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a source=*Value*	Specifies an existing **bosinst_data** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.

dump Resource

A **dump** resource represents a directory in which client dump directories are maintained. When this type of resource is allocated to a client, NIM creates a subdirectory for the client's exclusive use. This allocated subdirectory is initialized by the **dkls_init** or **dtls_init** operation, which creates an empty file in this subdirectory. After initialization, the client uses this file to store any **dump** images it creates.

> **Note:** If you subsequently deallocate this resource, NIM removes the **dump** file and the subdirectory NIM created for the client's use.

Defining a dump Resource

The command line syntax for defining a **dump** resource is:

```
nim -o define -t dump -a Attribute=Value ... DumpName
```

The following attributes are required for this resource:

-a location=*Value*	Specifies the full path name of the parent directory for the client **dump** files.
-a server=*Value*	Specifies the name of the machine where the directory for the **dump** resource will be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

exclude_files Resource

An **exclude_files** resource represents a file that contains a list of files and directories that should be excluded when creating a system backup image. This resource may be used when a **mksysb** resource is being created from a running NIM client.

Defining an exclude_files Resource

The command line syntax for defining an **exclude_files** resource is:

```
nim -o define -t exclude_files -a Attribute=Value ... exclude_filesName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the file containing the list of files and directories to exclude from the **mksysb**.

Required Attributes	Definition
-a server=*Value*	Specifies the name of the machine where the file for the **exclude_files** resource resides.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a source=*Value*	Specifies an existing **exclude_files** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.

fix_bundle Resource

A **fix_bundle** resource represents a file containing **fix** keywords to be used by the **instfix** command which is called by the NIM **cust** and **fix_query** operations. NIM mounts the **fix_bundle** resource on the client so it can be used by the local **instfix** command. NIM automatically unmounts the resource when the operation has completed.

A fix can include either a single fileset update or multiple fileset updates that are related in some way; fixes are identified by unique keywords. When a fix is identified with an Authorized Program Analysis Report (APAR) number, it includes all the fileset updates that are necessary to fix the reported software problem identified by that number.

Defining a fix_bundle Resource

The command line syntax for defining a **fix_bundle** resource is:

```
nim -o define -t fix_bundle -a Attribute=Value ... fix_bundleName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the file containing the list of fixes to manage.
-a server=*Value*	Specifies the name of the machine where the **fix_bundle** resource file resides.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

Optional Attributes	Definition
-a source=*Value*	Specifies an existing **fix_bundle** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.

home Resource

A **home** resource represents a directory in which client **/home** directories are maintained. When this type of resource is allocated to a client, NIM creates a subdirectory for the client's exclusive use. This allocated subdirectory is subsequently initialized when you perform the **dkls_init** or **dtls_init** operation. After initialization, any time the client performs a network boot, the client NFS mounts this subdirectory over **/home** to gain access to the **home** directory that has been set up for its use. This subdirectory remains mounted over **/home** on the client as long as the client is running.

> **Note:** Whenever this resource is deallocated, NIM removes the subdirectory that was created for the client's use. Therefore, back up any files you want to save in the client's subdirectory before you deallocate a resource of this type.

Defining a home Resource

The command line syntax for defining a **home** resource is:

```
nim -o define -t home -a Attribute=Value ... HomeName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the parent directory for the client **/home** directories.
-a server=*Value*	Specifies the name of the machine where the directory for the **home** resource will be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

image_data Resource

An **image_data** resource represents a file that contains information for the BOS install program. This information describes how physical disks and file systems should be configured in the root volume group during installation. Normally, the BOS install program determines default values that should be used, or uses an **image.data** file from a **mksysb** being restored. Only in special cases would you use a customized **image_data** resource.

A sample **image.data** file (*SPOT_Offset*/**usr/lpp/bosinst/image.template**) is located on the **SPOT** resource server. For more information about **image.data** files, see the *AIX 5L Version 5.1 Files Reference* and the *AIX 5L Version 5.1 Installation Guide*.

Defining an image_data Resource

The command line syntax for defining an **image_data** resource is:

```
nim -o define -t image_data -a Attribute=Value ... image_dataName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the **image_data** resource file.
-a server=*Value*	Specifies the name of the machine where the **image_data** resource file resides.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a source=*Value*	Specifies an existing **image_data** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.

installp_bundle Resource

An **installp_bundle** resource represents a file that contains the names of filesets that should be managed by NIM. During an installation or maintenance operation, NIM mounts the **installp_bundle** file on the client machine so it can be used by the local **installp** command. NIM automatically unmounts the resource from the client when the operation has completed.

Defining an installp_bundle Resource

The command line syntax for defining an **installp_bundle** resource is:

```
nim -o define -t installp_bundle -a Attribute=Value ... installp_bundleName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the file containing the list of software to manage.
-a server=*Value*	Specifies the name of the machine where the **installp_bundle** resource file resides.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a source=*Value*	Specifies an existing **installp_bundle** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.

lpp_source Resource

An **lpp_source** resource represents a directory in which software installation images are stored. If the **lpp_source** contains the minimum set of support images required to install a machine, it is given the **simages** attribute and can be used for BOS installation (**bos_inst**) operations. If an **lpp_source** does not contain enough software to be an **simages lpp_source**, then it can only be used in NIM **cust** operations to install software on running machines and **SPOTs**.

NIM uses an **lpp_source** for an installation operation by first mounting the **lpp_source** on the client machine. The **installp** commands are then started on the client using the mounted **lpp_source** as the source for installation images. When the installation operation has completed, NIM automatically unmounts the resource.

In addition to providing images to install machines, **lpp_source** resources can also be used to create and update **SPOT** resources.

The minimum set of images required for an **lpp_source** to have the **simages** attribute in AIX 4.2 and later are:

POWER-based	Itanium-based
bos	**bos**
bos.64bit (if AIX 4.3 or later)	N/A
bos.rte.up (if AIX 4.1)	N/A
bos.rte.mp (if AIX 4.1)	N/A
bos.up (if AIX 4.2 or later)	N/A
bos.mp (if AIX 4.2 or later)	**bos.mp**
bos.net	**bos.net**
bos.diag	**bos.diag**
bos.sysmgt	**bos.sysmgt**
bos.terminfo	**bos.terminfo**
bos.terminfo.data	**bos.terminfo.data**
devices.base	**devices.ia64.base**
devices.buc	N/A
devices.common	**devices.common**
devices.graphics	**devices.graphics**

devices.mca	**devices.pci**
devices.rs6ksmp.base	N/A
devices.scsi	**devices.scsi**
N/A	**devices.ide**
devices.sio	**devices.isa_sio**
devices.sys	N/A
devices.tty	**devices.tty**
xlC.rte	**xlC.rte**

Note: When copying device images to a directory that you plan to define as an **lpp_source**, be sure to copy all the device images for a given type of device. For example, `cp /cdfs/usr/sys/inst.images/devices.pci.* < lpp_source directory > ).`

When using the **packages** (page 91) attribute, to specify the desired images, you can use **installp** syntax, such as **devices.pci.all** to request all the available **pci** device images.

You can define an **lpp_source** in several ways:

- If a directory containing installation images already exists, it can be directly defined as an **lpp_source** resource.
- If a directory should be created and populated by NIM with the default set of support images for a BOS install, use the **source** attribute when defining the resource. This attribute specifies the name of the device that contains the installation images. NIM copies the software images from this device into the location specified for the **lpp_source**. The images copied will include those from the **simages** list, all available device support, and some additional software that is typically installed as well (for example, X11).
- If an **lpp_source** should be created from a source device using a list of software other than the default set of images, specify the **packages** attribute when defining the **lpp_source**. Use the **packages** attribute to list the alternative set of software images to copy.

The size of an **lpp_source** may vary greatly with the amount of software it includes. A minimum **lpp_source** with just enough software to qualify for the **simages** attribute may be under 100MB, but a default **lpp_source** created from a CD-ROM may be over 350MB. It is recommended that a separate file system be created to contain an **lpp_source** so the space can be more easily managed. By default, NIM automatically expands a file system as needed when creating an **lpp_source** and copying images from a source device.

Defining an lpp_source Resource

The command line syntax for defining an **lpp_source** resource is:

```
nim -o define -t lpp_source -a Attribute=Value ... lpp_sourceName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the directory that will contain the installation images.
-a server=*Value*	Specifies the name of the machine where the **lpp_source** is to be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the **lpp_source**.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a packages=*Value*	Specifies a list of filesets to copy into the **lpp_source** if the default list of images is not desired.
-a source=*Value*	Identifies the source device for copying installation images when defining the **lpp_source**. This attribute is not required if the location of the **lpp_source** already contains installation images.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

If a migration installation will be performed on NIM client machines, the **lpp_source** used in the operation must contain all the required software to migrate the machine.

If the directory specified in the **location** attribute does not exist, NIM will create the directory. NIM will also remove the directory and its contents if the **lpp_source** is later removed.

mksysb Resource

A **mksysb** resource represents a file that is a system backup image created using the **mksysb** command. This type of resource can be used as the source for the installation of a client. The **mksysb** image must reside on the hard disk of a machine in the NIM environment in order to be defined as a resource. It cannot be located on a tape or other external media.

A **mksysb** resource can be defined from an image that already exists on the hard disk of the NIM master or any NIM client. If such an image does not exist, it can be created when the resource is defined. To create the image when the resource is defined, specify the name of the NIM client that will be the **source** for the backup, and set the **mk_image** attribute to **yes** in the command to define the **mksysb** resource. Use an **exclude_files** resource to list any files and directories that should not be included in the backup image.

Defining a mksysb Resource

The command line syntax for defining a **mksysb** resource is:

```
nim -o define -t mksysb -a Attribute=Value ... mksysbName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the **mksysb** image.
-a server=*Value*	Specifies the name of the machine where the **mksysb** image resides or is to be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the **mksysb**.
-a exclude_files=*Value*	Specifies an **exclude_files** resource to use to exclude files and directories from the system backup.

Optional Attributes	Definition
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a mk_image=*Value*	Specifies the flag to use to create a **mksysb** image from a machine in the NIM environment.
-a mksysb_flags=*Value*	Specifies the flags to use to tell the command how to create the backup.
-a size_preview=*Value*	Specifies the flag to verify that space is available before creating a **mksysb** image.
-a source=*Value*	Specifies the name of the machine to be backed up in the **mksysb** image. NIM clients must be running AIX 4.2 or later to be valid source machines.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a source=*Value*	Specifies the name of the machine to be backed up in the mksysb image if the **mk_image** attribute is specified. NIM clients must be running AIX 4.2 or later to be valid source machines. If the **mk_image** attribute is not specified then this value specifies an existing mksysb resource to be replicated when defining a new mksysb resource. The file pointed to by the source resource will be copied to the new location.

nim_script Resource

The **nim_script** resource is an internally managed NIM resource used to indicate that a script should be run by NIM as part of a NIM operation. The **nim_script** resource is automatically allocated to support some NIM operations, and it is automatically deallocated when the operations complete.

Depending on the operation, NIM will use the following rules to determine which NIM server to place the **nim_script resource** on:

- for a **bos_inst** operation the **nim_script** resource will be placed on the **SPOT** server.
- for **cust** operation with a **lpp_source** the **nim_script** resource will be placed on the **lpp_source** server.
- for a **cust** operataion without a **lpp_source** the **nim_script** resource will be placed on the script server.
- otherwise it will be placed on the NIM master.

paging Resource

A **paging** resource represents a directory where client paging files are maintained. When this type of resource is allocated to a client, NIM creates a subdirectory for the client's exclusive use. This allocated subdirectory is initialized by the **dkls_init** or **dtls_init** operation, which creates a file in this subdirectory that the client configures as a paging device when it performs a network boot. By default, 32MB are reserved for this file. A different value can be specified using the **size** flag when the **dkls_init** or **dtls_init** operation is performed.

After this resource has been initialized for a client, it is configured as a paging device by the client each time the client performs a network boot.

> **Note:** If you subsequently deallocate this resource, NIM removes the **paging** file and the subdirectory it created for the client's use.

Defining a paging Resource

The command line syntax for defining a **paging** resource is:

```
nim -o define -t paging -a Attribute=Value ... PagingName
```

The following attributes are required for this resource:

-a location=*Value*	Specifies the full path name of the parent directory for the client **paging** files.
-a server=*Value*	Specifies the name of the machine where the directory for the **paging** resource will be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

resolv_conf Resource

A **resolv_conf** resource represents a file containing valid **/etc/resolv.conf** entries which define Domain Name Protocol name-server information for local resolver routines. A **resolv_conf** resource can be allocated to a standalone machine as part of a **bos_inst** operation or to a diskless or dataless machine as part of a **dkls_init** or **dtls_init** operation. Upon successful installation and reboot, the machine will be configured to use the domain name services defined by the resource.

The following are sample entries in a **resolv_conf** resource file:

```
nameserver      129.35.143.253
domain          test.ibm.com
```

Defining a resolv_conf Resource

The command line syntax for defining a **resolv_conf** resource is:

```
nim -o define -t resolv_conf -a Attribute=Value ... resolv_confName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the file containing the information for domain name server (DNS) name resolution.
-a server=*Value*	Specifies the name of the machine where the **resolv_conf** resource file resides.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.

Optional Attributes	Definition
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a source=*Value*	Specifies an existing **resolv_conf** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.

root Resource

A **root** resource represents a directory in which client **root** directories are maintained. When this type of resource is allocated to a diskless or a dataless client, NIM creates a subdirectory for the client's exclusive use. This allocated subdirectory is subsequently initialized when you perform the **dkls_init** or **dtls_init** operation.

After initialization, anytime the client performs a network boot, the client NFS mounts this subdirectory over *"/"* to gain access to the **root** directory that has been set up for its use. This subdirectory remains mounted over *"/"* on the client as long as the client is running.

> **Note:** Whenever this resource is deallocated, NIM removes the subdirectory that was created for the client's use. Therefore, any files you want to save in the client's subdirectory should be backed up before you deallocate a resource of this type.

Defining a root Resource

The command line syntax for defining a **root** resource is:

```
nim -o define -t root -a Attribute=Value ... RootName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the directory under which client **root** directories will be created.
-a server=*Value*	Specifies the name of the machine where the directory for the **root** resource will be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

script Resource

A **script** resource represents a file that is a user-defined shell script. Once defined, this type of resource can be used to perform processing on a client as part of a NIM **cust** or **bos_inst** operation.

script resources are always run by NIM after software installation is performed in **cust** or **bos_inst** operations. This allows the scripts to perform configuration processing on the client after all the software is installed. Multiple **script** resources can be allocated for client use, but the order that the scripts will be run is not predictable.

> **Note: script** resources must not point to files that reside in the **/export/nim/scripts** directory. This directory is used for the **nim_script** resource that is managed by NIM. NFS restrictions prevent defining multiple resources in the same location.

Defining a script Resource

The command line syntax for defining a **script** resource is:

```
nim -o define -t script -a Attribute=Value ... ScriptName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the **script** resource file.
-a server=*Value*	Specifies the name of the machine where the **script** resource file resides.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a source=*Value*	Specifies an existing **script** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.

shared_home Resource

A **shared_home** resource represents a directory that can be used as a common **/home** directory by one or more clients. When this type of resource is allocated to a client, and when the **dkls_init** or **dtls_init** operation is performed, NIM configures the client's configuration to use this common directory. After initialization, anytime the client performs a network boot, the client NFS mounts this common directory over its **/home** directory. This common directory remains mounted as long as the client is running.

> **Note:** Whenever this resource is deallocated, NIM only changes the client's configuration so that this directory is no longer used by the client. NIM does not remove the common directory.

Defining a shared_home Resource

The command line syntax for defining a **shared_home** resource is:

```
nim -o define -t shared_home -a Attribute=Value ... shared_homeName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the directory to be used as a common **/home** directory among clients.
-a server=*Value*	Specifies the name of the machine where the directory for the **shared_home** resource will be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

SPOT (Shared Product Object Tree) Resource

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

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 **/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 those that are 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**. The second method, creating a non-**/usr SPOT**, 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 **SPOT**s, all with different additional packages and filesets installed, serving different clients.

Note that you should not create a non-**/usr SPOT** in a subdirectory of **/usr**.

A **SPOT** varies in size from 100MB up to, and sometimes in excess of, 300MB 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**

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 4MB 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.

The Micro Channel-based systems support booting from the network using Token-Ring, Ethernet, or FDDI. The POWER-based 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.

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.

To list the software installed in a **SPOT**, enter the following command:

```
nim -o lslpp SPOTName
```

If you want to change your **/usr SPOT** back to a normal **/usr** file system, you must remove the **SPOT** from the NIM database.

For information about software installation and maintenance tasks you can perform on a **SPOT**, see the following sections:

- "Customizing NIM Clients and SPOT Resources" on page 23
- "Maintaining Software on Standalone Clients and SPOT Resources" on page 44

Network Boot Images for AIX 4.1 SPOTs

Each network boot image supports a single network type and a single platform type. The network boot image files are named *SPOTname.Platform.Network*. The network types are Token-Ring, Ethernet, and FDDI. Network boot capability is not provided over NIM's generic network type. The platform types are:

rs6k	Used for Micro Channel-based, uniprocessor machines
rs6ksmp	Used for Micro Channel-based, symmetric multiprocessor machines
rspc	Used for PCI bus-based, uniprocessor machines
rspcsmp	Used for PCI bus-based, symmetric multiprocessor machines

A total of seven network boot images are created for different combinations of platforms and network interfaces. The network boot images located in **/tftpboot** for a **SPOT** named `41spot`, look similar to the following:

```
41spot.rs6k.ent

41spot.rs6k.fddi

41spot.rs6k.tok

41spot.rs6ksmp.ent

41spot.rs6ksmp.tok

41spot.rspc.ent

41spot.rspc.tok
```

Network Boot Images for AIX 4.2 SPOTs

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 family/POWER family2/P2SC/POWER-based MCA bus-based machines.
rspc	Used for POWER-based Reference Platform (PREP) Architecture-based machines.
chrp	Used for POWER-based Common Hardware Reference Platform (CHRP) Architecture-based machines.

The **rs6ksmp** platform for AIX 4.2 (and later) **SPOTs** is 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** for a **SPOT** named `42spot` look similar to the following:

```
42spot.rs6k.mp.ent

42spot.rs6k.mp.fddi

42spot.rs6k.mp.tok

42spot.rs6k.up.ent
```

```
42spot.rs6k.up.fddi

42spot.rs6k.up.tok

42spot.rspc.mp.ent

42spot.rspc.mp.tok

42spot.rspc.up.ent

42spot.rspc.up.tok
```

The amount of space used in the **/tftpboot** directory for boot images may become very large. An AIX 4.2.1 (or later) **SPOT** that supports network boot for all possible combinations of platforms, kernel types, and network adapters may require as much as 60MB 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.

Network Boot Images for AIX 4.3 SPOTs

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. See “Creating Network Boot Images to Support Only the Defined Clients and Networks” on page 70 for further information.

Defining a SPOT Resource

The command line syntax for defining a **SPOT** resource is:

```
nim -o define -t spot -a Attribute=Value ... SPOTName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the parent directory under which the **SPOT** is to be created.
-a server=*Value*	Specifies the name of the machine where the **SPOT** is to be created.
-a source=*Value*	Identifies the source device for installation images to create and install the **SPOT**.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a auto_expand=*Value*	Expands the file system as needed when installing the **SPOT**. The default value is **yes**.
-a comments=*Value*	Describes the **SPOT**.
-a debug=*Value*	Builds debug-enabled network boot images. The default value is **no**.
-a installp_flags=*Value*	Specifies the flags that describe how **installp** should install software into the **SPOT**. The default value is **agQX**.
-a show_progress=*Value*	Shows **installp** output as **SPOT** is installed. The default value is **yes**.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

Note: The creation of a **SPOT**, by default, produces a large amount of output. Be sure to scan through the output to look for nonfatal errors and warnings that may not be evident from a successful return code.

tmp Resource

A **tmp** resource represents a directory where client **/tmp** files are maintained. When this type of resource is allocated to a client, NIM creates a subdirectory for the client's exclusive use. This allocated subdirectory is subsequently initialized when you perform the **dkls_init** or **dtls_init** operation. After initialization, anytime the client performs a network boot, the client NFS mounts this subdirectory over **/tmp** to gain access to the **/tmp** directory that has been set up for its use. This subdirectory remains mounted over **/tmp** on the client as long as the client is running.

> **Note:** Whenever this resource is deallocated, NIM removes the subdirectory that was created for the client's use. Therefore, back up any files you want to save in the client's subdirectory before you deallocate a resource of this type.

Defining a tmp Resource

The command line syntax for defining a **tmp** resource is:

```
nim -o define -t tmp -a Attribute=Value ... TmpName
```

The following attributes are required for this resource:

Required Attributes	Definition
-a location=*Value*	Specifies the full path name of the directory where client**/tmp** directories will be created.
-a server=*Value*	Specifies the name of the machine where the directory for the **tmp** resource will be created.

The following attributes are optional for this resource:

Optional Attributes	Definition
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.

Distributed NIM Resources

Usually, a NIM administrator will use the NIM master as the server for all resources. This strategy keeps all resources together on one machine. However, there are several reasons to distribute resources onto client machines:

- If the NIM environment requires several large resources to be defined, it may not be possible to put them all on the same server because of disk space limitations. Creating resources on different machines allows the burden of disk consumption to be distributed over several machines.
- Serving resources from different machines helps avoid bottlenecks when performing NIM operations on large numbers of clients. Bottlenecks can occur on server machines or on network gateways, so it may be beneficial to distribute resources across servers running in different subnets.
- Multiple resources of the same type can be created on different machines to increase the availability of resources when servers are taken offline for scheduled maintenance.

- Some **SPOT** resources at certain levels cannot be served by some machines at certain levels. Specifically, **SPOT** creation is not supported when the level of AIX installed in the **SPOT** is higher than the level of AIX running on the server. When you are creating **SPOTs** at multiple levels, it may be necessary to distribute the **SPOTs** on different servers.

Distributing resources on different machines in the NIM environment is simply a matter of specifying the correct server information when the resource is defined. After the resources are created, they are used no differently than resources defined on the master.

NIM Operations

A large number of operations can be performed to manage a NIM environment and perform software installation and maintenance. The Web-based System Manager and SMIT interfaces are designed to hide much of the detail required for the command line interface. Therefore, this section only documents the operations for the command line. All of this information applies to the other interfaces as well, but discussion of those interfaces is deferred to the online contextual help available for those applications.

Most NIM operations are performed by running the **nim** command with various attributes for each possible operation. The command line syntax is as follows:

```
nim -o OperationName -a Attribute=Value ... TargetName
```

The NIM operations you can perform are:

“allocate”	“diag” on page 111	“remove” on page 116
“alt_disk_install” on page 102	“dkls_init” on page 112	“reset” on page 116
“bos_inst” on page 104	“dtls_init” on page 113	“select” on page 117
“change” on page 108	“fix_query” on page 113	“showlog” on page 117
“check” on page 109	“lppchk” on page 114	“showres” on page 118
“cust” on page 109	“maint” on page 115	“sync_roots” on page 119
“deallocate” on page 111	“maint_boot” on page 115	“unconfig” on page 120
“define” on page 111	“reboot” on page 116	

allocate

The **allocate** operation is used to make resources available to NIM clients for subsequent operations. In AIX 4.2 and later, it is generally unnecessary to perform explicit allocations prior to NIM operations, because the operations accept resource names as attributes to the command.

The command line syntax for the **allocate** operation is as follows:

```
nim -o allocate -a ResourceType=ResourceName ... TargetName
```

The target of an **allocate** operation may be a NIM client or group of NIM clients.

The following attribute can be specified for the **allocate** operation:

-a *ResourceType=ResourceName* (required)	Specifies the resource to allocate to the client, for example, `lpp_source=42_images`.

When a resource is allocated to a client, an entry is added to the **/etc/exports** file on the resource server to NFS export the resource to the client. The allocation count for the resource is also incremented. When the allocation count is greater than 0, the resource cannot be modified. During NIM operations, a client mounts and uses the resources that have been allocated to it.

alt_disk_install

The **alt_disk_install** operation (available in AIX 4.3) can be used to install a **mksysb** image on a client system's alternate disk or disks, or it can be used to clone a client running **rootvg** to an alternate disk.

The command line syntax for the **alt_disk_install mksysb** operation is as follows:

```
nim -o alt_disk_install -a source=mksysb -a mksysb=mksysb_resource
 \-a disk=target_disk(s) -a attribute=Value....  TargetName
```

The command line syntax for the **alt_disk_install rootvg** clone operation is as follows:

```
nim -o alt_disk_install -a source=rootvg -a disk=target_disk(s)
 \-a attribute=Value....  TargetName
```

The target of an **alt_disk_install** operation can be a standalone NIM client or a group of standalone NIM clients. The clients must be running AIX 4.1.4 or higher and have the **bos.alt_disk_install.rte** fileset installed.

To display the alternate disk installation status while the installation is progressing, enter the following command on the master:

```
lsnim -a info -a Cstate ClientName
```

OR

```
lsnim -l ClientName
```

The following are required attributes for **alt_disk_install mksysb** operation:

-a source=mksysb	Specifies the type of **alt_disk_install** to perform.
-a disk=*target_disk(s)*	Specifies the disks on the client system that the **mksysb** image will be restored. This disk or these disks must not currently contain any volume group definition. The **lspv** command should show these disks as belonging to volume group **None**. If you are specifying more than one disk, the disk names must be enclosed in a set of single quotes; for example, `'hdisk2 hdisk3'`.
-a mksysb=*mksysb_resource*	Specifies the **mksysb** resource to use. Currently, only AIX 4.3 **mksysb** images are supported by the **alt_disk_install** command, but they can be installed on AIX 4.1.4 and higher systems.

The following are required attributes for the **alt_disk_install rootvg** clone operation:

-a source=rootvg	Specifies the type of **alt_disk_install** to perform.
-a disk=*target_disk(s)*	Specifies the disks on the client system that the **mksysb** image will be restored. This disk or these disks must not currently contain any volume group definition. The **lspv** command should show these disks as belonging to volume group **None**. If you are specifying more than one disk, the disk names must be enclosed in a set of single quotes; for example, `'hdisk2 hdisk3'`.

The following are optional attributes that can be specified for both **alt_disk_install mksysb** and the **alt_disk_install rootvg** clone operation:

-a concurrent=Value	Specifies the maximum number of machines from the selected group that should be installing at any given time. This attribute is only valid when the target of the operation is a machine group. If specified, NIM will monitor the progress of all machines in the group and attempt to keep no more or less than the number specified installing until all machines in the group are installed.
-a set_bootlist=*Value*	Specifies whether to set the bootlist to point to the new **rootvg** when the install is complete. *Value* can be `yes` or `no`, where `yes` is the default value. The next time the system is rebooted, it will boot from the newly installed alternate disk if *Value* is set to `yes`.
-a boot_client=*Value*	Specifies whether to reboot the client when the **alt_disk_install** operation is completed. *Value* can be `yes` or `no`, where `no` is the default value. This attribute would normally be set only if the **set_bootlist** attribute was also set to `yes`.
-a debug=*Value*	Specifies whether to print debug **(set -x)** output from the **alt_disk_install** script. *Value* can be `yes` or `no`, where `no` is the default value. This output does not go to the screen, but is saved to the NIM log, **/var/adm/ras/nim.alt_disk_install**, on the client system. This file can be checked after the **alt_disk_install** has completed.
-a image_data=*Value*	Specifies the image_data resource to use when creating the new alternate **rootvg** and its logical volumes and file systems. The new volume group created must be large enough to restore the **mksysb** image or a copy of the running **rootvg**. An **exclude_files** attribute can also be used with an **alt_disk_install rootvg** clone to specify files or directories that should not be backed up.
-a resolv_conf=*Value*	Specifies the **resolv_conf** resource to use for configuring the domain and name resolution on the client system when the system is rebooted. This is the **/etc/resolv_conf** file that will be copied into the alternate disk's file system. This may be useful if the **mksysb** image you are using has a different **/etc/resolv_conf** file than the one you want the client to retain.
-a script=*Value*	Specifies the script resource to call at the end of the **alt_disk_install** operation. This script is called on the running system before the **/alt_inst** file systems are unmounted, so files can be copied from the running system to the **/alt_inst** file systems before the reboot. This is the only opportunity to copy or modify files in the alternate file system because the logical volume names will be changed to match those of **rootvg**, and they will not be accessible until the system is rebooted with the new alternate **rootvg**.
-a time_limit=Value,	Specifies the maximum number of hours that should elapse before ceasing to initiate installation of additional members of the selected group of machines. This value can only be specified when limiting the number of concurrent operations on a group.
-a verbose=*Value*	Specifies whether to show files as they are being backed up for a **rootvg** clone, or to show files as they are being restored for a **mksysb** install. *Value* can be `yes` or `no`, where `no` is the default value. The output goes to the **alt_disk_install** log on the client, **/var/adm/ras/alt_disk_inst.log**.

The following are optional attributes that can be specified only for the **alt_disk_install rootvg** clone operation:

-a exclude_files=*Value*	Specifies an **exclude_files** resource to use to exclude files and directories from the **rootvg**. Files and directories specified in this file will not be copied to the new cloned **rootvg**.

-a filesets=*Value*	Specifies the list of filesets to install into the alternate **rootvg** after the clone of the **rootvg** is complete.
-a fixes=*Value*	Specifies the APARs to install into the alternate **rootvg** after the clone of the running **rootvg**. The fixes are in the format "IX123456" or "update_all".
-a fix_bundle=*Value*	Specifies the **fix_bundle** resource that lists the APARs to install into the alternate **rootvg** after the clone of the running **rootvg**.
-a installp_bundle=*Value*	Specifies an **installp_bundle** resource that lists filesets to install into the alternate **rootvg** after the clone of the running **rootvg**.
-a installp_flags=*Value*	Tells **installp** how to apply the filesets, **installp_bundle**, fixes, or **fix_bundles** attributes. The default value is installp_flags=-acgX.

bos_inst

The **bos_inst** operation is used to install the AIX Base Operating System on standalone clients.

The command line syntax for the **bos_inst** operation is as follows:

```
nim -o bos_inst -a source=Value -a Attribute=Value ... TargetName
```

The target of a **bos_inst** operation can be a standalone NIM client or a group of standalone NIM clients.

The following are required attributes that can be specified for the **bos_inst** operation to install and customize a machine:

-a lpp_source=*Value*	Identifies the **lpp_source** resource to be used. The **lpp_source** specified must have the **simages** attribute. The **lpp_source** provides software for machine customization. It also provides the BOS image for installation if the **source** attribute is **rte**.
-a source=*Value*	Identifies the source for BOS run-time files. Valid values are: **rte** Installs from a BOS image in the **lpp_source**. **spot** Copies BOS run-time files from the **SPOT**. **mksysb** Installs the machine from a **mksysb** image.
-a spot=*Value*	Identifies the **SPOT** resource to be used. The **SPOT** provides support for network boot and operations in the boot environment. It also provides the BOS run-time files if the **source** attribute is **spot**.

The following are optional attributes that can be specified for the **bos_inst** operation:

-a accept_licenses=*Value*	Specifies whether license agreements should be accepted during BOS installation. This value must be set to **yes** or the ACCEPT_LICENSES field in the **bosinst_data** resource must be set to **yes**, before the installation process can complete. The default value is **accept_licenses=no**.

-a async=*Value*	Specifies whether NIM should perform operations on group members asynchronously and not wait for the operation to complete on one member before beginning the operation on the next. The default value is **async=yes**.
-a auto_expand=*Value*	Indicates whether or not to expand file systems when setting up a client for a **force_push** installation. The default value is **auto_expand=yes**.
-a boot_client=*Value*	Indicates whether or not NIM should attempt to reboot the client immediately for BOS installation. The **boot_client** attribute is the converse of the **no_client_boot** attribute. The default value is **boot_client=yes**, indicating that NIM should attempt to reboot the client.
-a bosinst_data=*Value*	Specifies the **bosinst_data** resource to use for non-prompted installation.
-a concurrent=Value	Specifies the maximum number of machines from the selected group that should be installing at any given time. This attribute is only valid when the target of the operation is a machine group. If specified, NIM will monitor the progress of all machines in the group and attempt to keep no more or less than the number specified installing until all machines in the group are installed.
-a filesets=*Value*	Specifies a list of filesets to install on the target after BOS installation.
-a force_push=*Value*	Indicates whether or not a **force_push** installation should occur. A **force_push** should be used for installing machines that are running, but are not configured with the NIM client fileset. See the “force_push Attribute” on page 108 for more information.
-a group=*Value*	Specifies the name of a resource group to use for installation. A resource group can be specified as an alternative to specifying multiple resources as separate attributes. If a resource group is specified, and it contains a **SPOT** and **lpp_source**, the **spot** and **lpp_source** attributes are no longer required.
-a image_data=*Value*	Specifies an **image_data** resource to describe how physical and logical data is organized on the client.
-a installp_bundle=*Value*	Specifies an **installp_bundle** resource that lists filesets to install on the target after BOS installation.
-a installp_flags=*Value*	Tells **installp** how to apply the filesets specified by the **filesets** or **installp_bundle** attributes. The default value is **installp_flags=-agQX**.
-a mksysb=*Value*	Provides the run-time files for BOS and other filesets if the **source** attribute is **mksysb**. The level of BOS run-time files in the **mksysb** must be equal to the level of the **SPOT** used for the installation.
-a no_client_boot=*Value*	Indicates whether or not NIM should attempt to reboot the client immediately for BOS installation. The **no_client_boot** attribute is the converse of the **boot_client** attribute. The default value is **no_client_boot=no**, indicating that NIM should attempt to reboot the client.
-a no_nim_client=*Value*	Indicates whether the target should remain in the NIM environment after installation completes. The default value is **no**, indicating that the target system should remain in the NIM environment.
-a preserve_res=*Value*	Indicates whether or not resources in non-rootvg file systems should be preserved on the client system being installed. The default value is **preserve_res=no**.

-a resolv_conf=*Value*	Specifies the **resolv_conf** resource to use for configuring domain and name resolution on a client.
-a script=*Value*	Specifies the **script** resource to be run on the target system after all software has been installed.
-a set_bootlist=*Value*	Indicates whether or not NIM should set the bootlist of the client so that the client boots over the network on the next reboot. Usually, **set_bootlist** would be **yes** if the client is not going to be rebooted immediately for installation (**no_client_boot=yes** or **boot_client=no**). The default value is **set_bootlist=no**.
-a show_progress=*Value*	Indicates whether status should be displayed for each group member when the installation target is a group of machines. The default value is **show_progress=yes**.
-a time_limit=Value,	Specifies the maximum number of hours that should elapse before ceasing to initiate installation of additional members of the selected group of machines. This value can only be specified when limiting the number of concurrent operations on a group.
-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

When a **bos_inst** operation is performed on a client, several things occur.

On the **SPOT** server:

1. A link is created in **/tftpboot** to a boot image matching the platform type, kernel type, and network adapter of the client.
2. The **/etc/bootptab** file is updated with client information to allow the boot image to be used.
3. A *ClientName***.info** file is created in **/tftpboot** to provide client-specific installation and configuration information in the boot environment.
4. The **/etc/tftpaccess.ctl** file is modified, if necessary, to allow access to the **/tftpboot** directory.

On the target system:

1. The bootlist is modified so the network adapter is the default boot device for normal mode boot, unless **no_client_boot=yes**, **set_bootlist=no**, and **force_push=no** are specified.
2. The client is rebooted to begin the installation, unless **no_client_boot=yes**, **boot_client=no**, and **force_push=no** are specified.

When the client boots over the network adapter, it obtains the boot image from the **SPOT** server. The boot image configures devices and sets up the machine for the BOS install. The *Client*.info file is transferred to the client machine; and based on its contents, the network adapter is configured, routes are added, and NIM resources are mounted in the boot environment. Processing control is then passed to the BOS install program.

NIM BOS Installation Details

The BOS installation program requires access to an image that contains the BOS run-time files. This image is used by the BOS installation program to populate the target's **/usr** file system. In the NIM environment, this image can come from one of the three following resources:

- A BOS run-time image that is part of the **lpp_source** resource that has been allocated to the target.
- A **SPOT** resource that has been allocated to the target.
- A **mksysb** image that has been allocated to the target.

Because a **spot** and **lpp_source** are always required to support the **bos_inst** operation, there are always at least two different kinds of BOS images available to the target. You can optionally create a **mksysb** image for your target machine. However, only one source can be used for BOS installation.

To indicate which BOS image to use, you need to specify the **source** attribute when performing the **bos_inst** operation. The **source** attribute may have one of the following values:

rte	When an **rte** value (the default) is used for the **source** attribute, NIM directs the BOS installation program to use the BOS run-time image that is in the **lpp_source** directory. This image contains only the BOS run-time files; it does not contain any optional software packages. By installing only the BOS run-time files, the installed target may have more free disk space than if the **SPOT** source type is used. However, selecting an **rte** source may increase the BOS installation time, because the BOS installation program installs the appropriate device support after populating the target's **/usr** file system to make the target viable. The installation time may also be increased due to additional **installp** activity during the NIM customization phase. **Note:** The **rte** source must be used when performing BOS migration installation.
spot	In contrast to **rte**, choosing **spot** as the source for the BOS image results in a target with a **/usr** file system that has all the files that exist in the **SPOT**, because the BOS installation program copies all of the files from the **SPOT** into the target's **/usr** file system. A **SPOT** has many optional products already installed in it, and time may be saved when the **installp** command is executing during the BOS installation process. However, a **SPOT** is usually large; and if the target does not have enough disk space to accommodate the size of the **SPOT**, the installation operation fails.
mksysb	Using **mksysb** as the source results in a target that has the same configuration as the machine from which the **mksysb** image was created. This may save install and configuration time. Like a **spot** source, however, **mksysb** images could be very large, and the installation will fail if the target does not have enough disk space to accommodate the image.

After the installation is initiated from the master, the NIM master attempts to contact the target and execute a script that will force the system to reboot. The target system issues a BOOTP request to the server after it has shut down. The **bos_inst** operation is considered complete even if the target does not immediately issue a BOOTP request. The target must issue a BOOTP request to load a network boot image from the server to start the installation.

If the master is unable to contact the target system for any reason (for example, the system is turned off, it is not a running NIM client, or there is a network problem), a warning message is displayed and user intervention is then required at the target to issue the BOOTP request using the IPL ROM. See "Booting a Machine Over the Network" on page 159.

By default (`no_nim_client=no`), NIM also includes the customization required for the target to remain a NIM client after the install. This customization includes the installation and configuration of the **bos.sysmgt.nim.client** fileset and its requisite filesets, **bos.net.tcp.client** and **bos.net.nfs.client**, so that the NIM master can communicate with and control the client after installation. The **installp_flags** are passed to the **installp** command for installing the software on the standalone client. The **filesets** attribute can be used to install a list of additional filesets or software packages from the allocated **lpp_source**.

To display BOS installation status information while the installation is progressing, enter the following command on the master:

```
lsnim -a info -a Cstate ClientName
```

OR

```
lsnim -l ClientName
```

Errors in the allocation of a **nim_script** or **boot** resource type are fatal errors because the network BOS installation process cannot proceed without them. On the other hand, any error encountered during the attempt to cause the target to issue a BOOTP request is a nonfatal error to NIM because-at that point-NIM

has successfully initialized the environment to perform a network installation. As soon as the target has successfully loaded its allocated network boot image, the BOS installation process begins.

force_push Attribute

When assigned a value of **yes**, the **force_push** attribute tells NIM that the target of the **bos_inst** operation does not necessarily have the **bos.sysmgt.nim.client** fileset installed and configured. NIM will attempt to NFS mount or copy the minimal client support to the target system to perform an unattended installation or migration of the base operating system. If client support is copied to the target machine, NIM will automatically expand the necessary file systems on the target unless the **auto_expand** attribute to **bos_inst** is set to **no**. The **force_push** attribute requires that the client grant root **rsh** permissions to the master and that the key on the client be in the normal position. The **force_push** attribute also requires that a **bosinst_data** file be allocated to the target machine to indicate that a no-prompt installation should occur. The **force_push** attribute is set to **yes** by setting the Force Unattended Installation Enablement? option to **yes** when using SMIT to perform the **bos_inst** operation.

boot_client Attribute

When assigned a value of **no**, the **boot_client** attribute is used to instruct NIM not to attempt to initiate the BOS installation on the target machine after setting up the installation with the **bos_inst** operation. This allows a BOS installation to be set up while deferring the actual installation until the client is rebooted at a later time. Also, if the client is not a running machine, this attribute will avoid waiting for the reboot attempt to time-out or fail. If the installation of the client system is going to be initiated later from the server, the normal mode boot device list on the client must be set so that a network boot is attempted when the client is rebooted. No attempt is made to modify the boot list when **boot_client** is set to **no** unless the **force_push** or **set_bootlist** attributes are specified and set to a value of **yes**. The **boot_client** attribute is set to **no** by setting Initiate Boot Operation on Client to **no** when using SMIT to perform the **bos_inst** operation.

set_bootlist Attribute

The **set_bootlist** attribute can be used with the **boot_client** attribute to modify the boot device list on the client for normal mode so a network boot is attempted when the client is rebooted. It is not necessary to specify the **set_bootlist** attribute if the **force_push** attribute is set to **yes** or if **boot_client** is unspecified or set to **yes**. In both instances, the boot list will be modified as the default. The only valid values for **set_bootlist** are **yes** and **no**. In order to use the **set_bootlist** attribute on **bos_inst** operations, the **bos.sysmgt.nim.client** fileset must be installed at Version 4.1.3.0 or later on target clients. The **set_bootlist** attribute is set to **yes** by setting Set Boot List if Boot not Initiated on Client? when using SMIT to perform the **bos_inst** operation.

preserve_res Attribute

The **preserve_res** attribute can be used to preserve the NIM database definitions for resources residing on a NIM client that is being reinstalled. When set to **yes**, any resources that reside in file systems which are being preserved by the BOS installation process will also be preserved.

change

The **change** operation is used to modify attributes of NIM objects. The command line syntax is as follows:

```
nim -F -o change -a Attribute=Value ... TargetName
```

-F (optional)	Tells NIM to **force** the operation if the target is currently in use. The target of a **change** operation can be any network, machine, resource, or group in the NIM environment. Not all attributes can be modified on targets. Usually, the attributes are changed automatically as parts of other operations, so there is little need for you to use the **change** operation explicitly.

check

The **check** operation is used to verify the usability of a machine or resource in the NIM environment.

The command line syntax for the **check** operation is as follows:

```
nim -F -o check -a debug=Value TargetName
```

The target of a **check** operation can be any NIM client, a group of NIM clients, a **SPOT** resource, or an **lpp_source** resource.

The following list includes all the flags and attributes that can be specified for the **check** operation:

-F (optional)	Tells NIM to ″force″ the operation if the target is currently in use. If the **-F** flag is specified when the target is a **SPOT** resource, it will force the **SPOT**'s network boot images to be rebuilt. The **-F** flag is typically not required when performing the **check** operation on client machines.
-a debug=*Value* (optional)	Builds a **SPOT**'s network boot images in debug mode if **debug=yes** is specified. This attribute is only valid if the target is a **SPOT** resource. The default value is **debug=no**. See "Producing Debug Output for NIM BOS Installations" on page 155 for more information on this attribute.

When applied to NIM clients, the **check** operation updates the machine state (**Mstate**) of the client. A ping test is performed to check whether or not the client is reachable. After the **check** operation is performed, the client's **Mstate** is set to either **running** or **not running**.

When applied to **SPOT** resources, the **check** operation performs root synchronization for diskless and dataless clients and rebuilds the **SPOT**'s network boot images, if necessary.

When applied to **lpp_source** resources, the **check** operation updates the table of contents (.toc) file in the **lpp_source** directory. It also determines whether or not all filesets are included in the resources to qualify the **lpp_source** for the **simages** attribute.

cust

The **cust** operation is used to install software filesets and updates on standalone clients and **SPOT** resources.

See "Customizing NIM Clients and SPOT Resources" on page 23 for information on performing a software customization of standalone NIM clients.

The command line syntax for the **cust** operation is as follows:

```
nim -o cust -a Attribute=Value ... TargetName
```

The target of a **cust** operation can be a standalone NIM client, a group of standalone NIM clients, or a **SPOT** resource.

The following are required attributes that can be specified for the **cust** operation:

-a filesets=*Value*	Specifies a list of filesets to install on the target. This attribute is required unless an **installp_bundle** is used for the operation.
-a installp_bundle=*Value*	Specifies an **installp_bundle** resource that lists filesets to install on the target. This attribute is required unless the **filesets** attribute is specified.
-a lpp_source=*Value*	Identifies the **lpp_source** resource that will provide the installation images for the **cust** operation.

The following are optional attributes that can be specified for the **cust** operation:

-a accept_licenses=*Value*	Specifies whether software licenses should be automatically accepted during installation. If `accept_licenses=yes`, the `-Y` flag is passed on the **installp** command and licenses are automatically accepted. If `accept_licenses=no`, license processing is controlled by the **installp_flags** attribute. The default value is **accept_licenses=no**.
-a async=*Value*	Specifies whether NIM should perform operations on group members asynchronously and not wait for the operation to complete on one member before beginning the operation on the next. The default value is **async=yes**.
-a concurrent=Value	Specifies the maximum number of machines from the selected group that should be installing at any given time. This attribute is only valid when the target of the operation is a machine group. If specified, NIM will monitor the progress of all machines in the group and attempt to keep no more or less than the number specified installing until all machines in the group are installed.
-a fix_bundle=*Value*	Contains a list of fixes to install on the target. Fixes should be listed in the **fix_bundle** resource by APAR number with one number per line.
-a fixes=*Value*	Identifies a list of fixes to install on the target. Fixes should be listed by APAR number. For example, `fixes="IX12345 IX54321"`.
-a group=*Value*	Specifies the name of a resource group to use for the installation. A resource group can be specified as an alternative to specifying multiple resources as separate attributes. If a resource group is specified, and it contains an **lpp_source**, the **lpp_source** attribute is no longer required.
-a installp_flags=*Value*	Identifies the flags that tell **installp** how to apply the filesets specified by the **filesets**, **installp_bundle**, **fixes**, and **fix_bundle** attributes. The default value is **installp_flags=agQX**.
-a resolv_conf=*Value*	Specifies a **resolv_conf** resource for configuring domain and name resolution on a client.
-a script=*Value*	Specifies a **script** resource to be run on the target system after all software has been installed.
-a show_progress=*Value*	Indicates whether status should be displayed as software is installed. The default value is **show_progress=yes**.
-a time_limit=Value,	Specifies the maximum number of hours that should elapse before ceasing to initiate installation of additional members of the selected group of machines. This value can only be specified when limiting the number of concurrent operations on a group.

deallocate

The **deallocate** operation is used to unlock and unexport resources when they are no longer needed by NIM clients. It is generally unnecessary to perform explicit deallocations after NIM operations, because upon successful completion, operations will automatically deallocate resources from the clients.

The command line syntax for the **deallocate** operation is as follows:

```
nim -o deallocate -a ResourceType=ResourceName ... -a subclass=all TargetName
```

The target of a **deallocate** operation may be a NIM client or group of NIM clients.

The following list includes all the attributes that can be specified for the **deallocate** operation:

-a *ResourceType=ResourceName*	Specifies the resource to deallocate from the client, for example, `lpp_source=42_images`. This attribute is required.
-a subclass=all	Specifies that all resources should be deallocated from the target. This attribute is optional.

When a resource is deallocated from a client, the **/etc/exports** file on the resource server is modified to unexport the resource from the client. The allocation count for the resource is also decremented.

define

Creates objects in the NIM environment. Networks, machines, and resources can be created using the **define** operation.

The command line syntax for the **define** operation is as follows:

```
nim -o define -t ObjectType -a Attribute=Value ... ObjectName
```

The attributes for the **define** operation vary for the different object types. For a complete description of the attributes required to define the various NIM objects, see “NIM Networks” on page 79, “NIM Machines” on page 73, “NIM Resources” on page 83, and “NIM Groups” on page 120.

diag

The **diag** operation is used to prepare resources for a client to be network-booted into diagnostics mode.

The command line syntax for the **diag** operation is as follows:

```
nim -o diag -a Attribute=Value ... TargetName
```

The target of a **diag** operation can be any standalone NIM client or group of standalone NIM clients.

The following are required attributes that can be specified for the **diag** operation:

-a spot=*Value*	Specifies the **SPOT** resource to be used to provide network boot and diagnostics support.

The following are optional attributes that can be specified for the **diag** operation:

-a group=*Value*	Specifies the name of a resource group to use for the operation. A resource group can be specified as an alternative to specifying multiple resources as separate attributes.
-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

dkls_init

The **dkls_init** operation is used to prepare resources for use by a diskless client.

The command line syntax for the **dkls_init** operation is as follows:

```
nim -o dkls_init -a Attribute=Value ... TargetName
```

The target of a **dkls_init** operation can be any diskless NIM client or group of diskless NIM clients.

The following are required attributes that can be specified for the **dkls_init** operation:

-a dump=*Value*	Specifies the **dump** resource that contains client dump files.
-a paging=*Value*	Specifies the **paging** resource that contains client paging files.
-a root=*Value*	Specifies the **root** resource that contains the client root (**/**) directories. The **root** resource must be served by the same machine that serves the **SPOT**.
-a spot=*Value*	Specifies the **SPOT** resource to be used to provide network boot support and the **/usr** file system for clients.

The following are optional attributes that can be specified for the **dkls_init** operation:

-a group=*Value*	Specifies the name of a resource group to use for the installation. A resource group can be specified as an alternative to specifying multiple resources as separate attributes.
-a home=*Value*	Specifies the **home** resource that contains client **/home** directories.
-a resolv_conf=*Value*	Specifies the **resolv_conf** resource to configure Domain Name Protocol name server information on the client.
-a shared_home=*Value*	Specifies the **shared_home** resource that contains a common **/home** directory for multiple clients.
-a size=*Value*	Specifies the size in megabytes for client paging files.
-a tmp=*Value*	Specifies the **tmp** resource that contains client **/tmp** directories.
-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

The **dkls_init** operation populates client directories and creates client paging and dump files. A network boot image is also allocated to the client. When the client boots over the network, it obtains the boot image and is configured to mount the remaining resources.

dtls_init

The **dtls_init** operation is used to prepare resources for use by a dataless client.

The command line syntax for the **dtls_init** operation is as follows:

```
nim -o dtls_init -a Attribute=Value ... TargetName
```

The target of a **dtls_init** operation can be any dataless NIM client or group of dataless NIM clients.

The following are required attributes that can be specified for the **dtls_init** operation:

-a dump=*Value*	Specifies the **dump** resource that contains client dump files.
-a spot=*Value*	Specifies the **SPOT** resource to be used to provide network boot support and the **/usr** file system for clients.
-a root=*Value*	Specifies the **root** resource that contains the client root (**/**) directories. The **root** resource must be served by the same machine that serves the **SPOT**.

The following are optional attributes that can be specified for the **dtls_init** operation:

-a paging=*Value*	Specifies the paging resource containing client paging files.
-a group=*Value*	Specifies the name of a resource group to use for the installation. A resource group can be specified as an alternative to specifying multiple resources as separate attributes.
-a home=*Value*	Specifies the **home** resource that contains client **/home** directories.
-a resolv_conf=*Value*	Specifies the **resolv_conf** resource to configure Domain Name Protocol name server information on the client.
-a shared_home=*Value*	Specifies the **shared_home** resource that contains a common **/home** directory for multiple clients.
-a size=*Value*	Specifies the size in megabytes for client paging files.
-a tmp=*Value*	Specifies the **tmp** resource that contains client **/tmp** directories.
-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

The **dtls_init** operation populates client directories and creates client paging and dump files. A network boot image is also allocated to the client. When the client boots over the network, it obtains the boot image and is configured to mount the remaining resources.

fix_query

The **fix_query** operation is used to display whether specified fixes are installed on a client machine or a **SPOT** resource.

The command line syntax for the **fix_query** operation is as follows:

```
nim -o fix_query -a Attribute=Value ... TargetName
```

The target of a **fix_query** operation can be any standalone NIM client, group of standalone NIM clients, or **SPOT** resource.

The following are optional attributes that can be specified for the **fix_query** operation:

-a fix_bundle=*Value*	Specifies a **fix_bundle** resource containing a list of fix keywords. This attribute is required unless the **fixes** attribute is specified for the operation.
-a fixes=*Value*	Specifies a list of keywords for the **fix_query** operation. Fix keywords are APAR numbers used to identify software updates that can span multiple filesets. This attribute is required unless a **fix_bundle** is used for the operation.
-a group=*Value*	Specifies the name of a resource group to use for the operation. A resource group can be specified as an alternative to specifying multiple resources as separate attributes.
-a fix_query_flags=*Value*	Tells the **fix_query** operation how to display information. Valid flags are those used by the **instfix** command.
-a show_progress=*Value*	Indicates whether status should be displayed as the operation is performed. The default value is **show_progress=yes**.

Note: There are no required attributes for the **fix_query** operation.

lppchk

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

The command line syntax for the **lppchk** operation is as follows:

```
nim -o lppchk -a Attribute=Value ... TargetName
```

The target of a **lppchk** operation can be any standalone NIM client, a group of standalone NIM clients, or a **SPOT** resource.

The following are optional attributes that can be specified for the **lppchk** operation:

-a async=*Value*	Specifies whether NIM should perform operations on group members asynchronously and not wait for the operation to complete on one member before beginning the operation on the next. The default value is **async=yes**.
-a filesets=*Value*	Specifies a list of filesets on the target on which the **lppchk** operation will be performed.
-a lppchk_flags=*Value*	Tells the **lppchk** command how to perform software verification.
-a show_progress=*Value*	Indicates whether status should be displayed as the operation is performed. The default value is **show_progress=yes**.
-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

Note: There are no required attributes for the **lppchk** operation.

maint

The **maint** operation is used to deinstall software filesets and commit and reject updates on standalone clients and **SPOT** resources.

The command line syntax for the **maint** operation is as follows:

```
nim -o maint -a Attribute=Value ... TargetName
```

The target of a **maint** operation can be a standalone NIM client, a group of standalone NIM clients, or a **SPOT** resource.

The following are required attributes that can be specified for the **maint** operation:

-a installp_flags=*Value*	Identifies the flags that tell **installp** what to do with the installed software.

The following are optional attributes that can be specified for the **maint** operation:

-a async=*Value*	Specifies whether NIM should perform operations on group members asynchronously and not wait for the operation to complete on one member before beginning the operation on the next. The default value is **async=yes**.
-a filesets=*Value*	Specifies a list of filesets to be maintained on the target.
-a group=*Value*	Specifies the name of a resource group to use for the operation. A resource group can be specified as an alternative to specifying multiple resources as separate attributes.
-a installp_bundle=*Value*	Specifies an **installp_bundle** resource that contains a list of filesets to be maintained on the target.
-a show_progress=*Value*	Indicates whether status should be displayed as maintenance is performed. The default value is **show_progress=yes**.

maint_boot

The **maint_boot** operation is used to prepare resources for a client to be network-booted into maintenance mode.

The command line syntax for the **maint_boot** operation is as follows:

```
nim -o maint_boot -a Attribute=Value ... TargetName
```

The target of a **maint_boot** operation can be any standalone NIM client or group of standalone NIM clients.

The following are required attributes that can be specified for the **maint_boot** operation:

-a spot=*Value*	Specifies the **SPOT** resource to be used to provide network boot and maintenance mode support.

The following are optional attributes that can be specified for the **maint_boot** operation:

-a group=*Value*	Specifies the name of a resource group to use for the operation.

-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

After the **maint_boot** operation is performed, the client must be rebooted over the network to load the network boot image and enter maintenance mode.

reboot

The **reboot** operation is used to reboot a NIM client machine.

The command line syntax for the **reboot** operation is as follows:

```
nim -o reboot -a Attribute=Value ... TargetName
```

The target of a **reboot** operation can be any standalone NIM client or group of standalone NIM clients.

The following are optional attributes that can be specified for the **reboot** operation:

-a inst_warning=*Value*	Indicates whether or not a warning should be displayed to warn users that the machine will be rebooted. The default value is **inst_warning=yes**.

Note: There are no required attributes for the **reboot** operation.

remove

The **remove** operation is used to remove objects from the NIM environment. The command line syntax is as follows:

```
nim -o remove ObjectName
```

The **remove** operation does not take any attributes. The target of this operation can be any network, machine, resource, or group in the NIM environment.

reset

The **reset** operation 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.

The command line syntax for the **reset** operation is as follows:

```
nim -F -o reset TargetName
```

The target of a **reset** operation can be any NIM client, a group of NIM clients, or a **SPOT** resource.

The following list includes all the flags and attributes that can be specified for the **reset** operation:

-F (optional)	Tells NIM to ″force″ the operation if the target is currently in use.

When applied to NIM clients, the **reset** operation updates the control state (**Cstate**) of the client. After the **reset** operation is performed, the client's **Cstate** is set to **ready**, and it is possible to perform NIM operations on the client. Although the **Cstate** of the client is reset by the operation, resources are not deallocated automatically. See the "deallocate" on page 111 operation for information on deallocating resources.

When applied to **SPOT** resources, the **reset** operation updates the resource state (**Rstate**) of the **SPOT**. After the **reset** operation is performed, the **SPOT**'s **Rstate** is set to **ready**, and you can use the **SPOT** in NIM operations.

select

The **select** operation is used to include and exclude group members from operations performed on the group.

The command line syntax for the **select** operation is as follows:

```
nim -o select -a Attribute=Value ... TargetName
```

The target of a **select** operation must be a group of NIM clients.

The following are optional attributes that can be specified for the **select** operation:

-a exclude=*Value*	Specifies the name of the group member to exclude from operations on the group.
-a exclude_all=*Value*	Indicates that all members of the group should be excluded from operations on the group. Valid values are **yes** and **no**.
-a include=*Value*	Specifies the name of the group member to include in operations on the group.
-a include_all=*Value*	Indicates that all members of the group should be included in operations on the group. Valid values are **yes** and **no**.
-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

To display the group members that are included and excluded from operations, use the **lsnim -g** *GroupName* command syntax.

showlog

The **showlog** operation is used to list software installed on a NIM client or **SPOT** resource.

The command line syntax for the **showlog** operation is as follows:

```
nim -o showlog -a Attribute=Value ... TargetName
```

The target of a **showlog** operation can be any standalone NIM client, a group of standalone NIM clients, or a **SPOT** resource.

The following are optional attributes that can be specified for the **lppchk** operation:

-a full_log=*Value*	Indicates whether the entire log is displayed or only the last entry. The default value is **full_log=no**.
-a log_type=*Value*	Specifies the type of log to display. The log types supported for both standalone clients and SPOT resources are: **boot** Machine's boot log **bosinst** Output from the BOS installation program **devinst** Output from the installation of key system and device-driver software **lppchk** Log of the output from the **lppchk** operation executed on a standalone NIM client **nimerr** Errors encountered during execution of the **nim** command **niminst** Output from the installation of user-specified software (including installation of NIM client software during a **bos_inst** operation) **script** Output from any configuration script resources allocated for a **bos_inst** operation.
-a show_progress=*Value*	Indicates whether status should be displayed as the operation is performed. The default value is **show_progress=yes**.
-a verbose=*Value*	Displays information for debugging. Valid values are 1-5. Use **verbose=5** to show maximum detail. The default is to show no debugging output.

Note: There are no required attributes for the **showlog** operation.

showres

The **showres** operation is used to display the contents of a resource. The contents displayed will be appropriate for the type of resource on which the operation is run.

The command line syntax for the **showres** operation is as follows:

```
nim -o showres -a Attribute=Value ... TargetName
```

The target of a **showres** operation may be a **SPOT**, **lpp_source**, **script**, **bosinst_data**, **image_data**, **installp_bundle**, **fix_bundle**, **resolv_conf**, or **exclude_files** resource.

The following are optional attributes that can be specified for the **showres** operation:

-a filesets=*Value*	Specifies a list of filesets for which information should be displayed. This attribute is only applicable to **lpp_source** and **SPOT** targets.
-a installp_flags=*Value*	Specifies flags that tell the **installp** command how to format the display of filesets. This attribute is only applicable to **lpp_source** and **SPOT** targets.
-a instfix_flags=*Value*	Specifies flags that tell the **instfix** command how to format the display of fixes. This attribute is only applicable to **lpp_source** targets.

-a lslpp_flags=*Value*	Specifies flags that tell the **lslpp** command how to format the display of installed software. This attribute is only applicable to **SPOT** targets.
-a reference=*Value*	Specifies a reference machine or **SPOT** resource for fileset comparison. This attribute is only applicable when the target of the operation is an **lpp_source**. Available filesets in the **lpp_source** are compared against installed filesets in the reference machine or **SPOT**. If the **showres** operation is performed from a NIM client, the **reference** attribute is automatically set to the name of the client.
-a resource=*Value*	Specifies the name of the resource whose contents should be displayed. This attribute is only necessary when the **showres** operation is performed from a NIM client.
-a sm_inst_flags=*Value*	Specifies flags that tell the **sm_inst** command how to format the display of filesets. This attribute is only applicable to **lpp_source** and **SPOT** targets. This attribute must be used in conjunction with the **reference** attribute and is normally used only within the SMIT application.

Note: There are no required attributes for the **showres** operation.

When the target of the **showres** operation is a **SPOT**, the list of filesets installed in the **SPOT** is displayed.

When the target of the **showres** operation is an **lpp_source**, the list of filesets contained in the **lpp_source** is displayed.

For all other resources that are valid targets for the **showres** operation, the character contents of the files are displayed.

sync_roots

The **sync_roots** operation is used to verify that diskless and dataless clients have the correct root files for the **SPOT** resource they use.

The command line syntax for the **sync_roots** operation is as follows:

```
nim -F -o sync_roots -a num_parallel_syncs=Value TargetName
```

The target of a **sync_roots** operation must be a **SPOT** resource.

The following list includes all the flags and attributes that can be specified for the **sync_roots** operation:

-a num_parallel_syncs=*Value*	Specifies the number of client root directories to simultaneously synchronize with the **SPOT**'s root files. Valid values are numeric. The default value is **num_parallel_syncs=5**. This attribute is optional.
-F	Specifies that NIM should **force** the operation. This flag is optional.

A **sync_roots** operation can be performed automatically when the **check** operation is performed on a **SPOT**.

unconfig

The **unconfig** operation is used to unconfigure the NIM master. The **unconfig** operation must be performed before the NIM master can be reconfigured or the NIM master fileset can be deinstalled.

> **Attention:** Performing the **unconfig** operation removes all information from the NIM database and should be used with caution.

The command line syntax for the **unconfig** operation is as follows:

```
nim -o unconfig master
```

The target of the **unconfig** operation must be the NIM master.

No attributes can be specified for the **unconfig** operation.

The **unconfig** operation completely unconfigures the NIM master by performing the following:

- Removes the **nimesis** and **nimd** daemon entries from the System Resource Controller (SRC).
- Removes all data from the **nim_attr** and **nim_object** databases.

NIM Groups

NIM groups are essentially lists of machines and resources. Groups can be used in NIM operations to simplify repetitive administrative tasks.

Machine Groups

Machine groups are used to represent collections of machines of similar types. The machine types in a group must all be the same (for example, standalone, diskless, or dataless), because some NIM operations are restricted to certain target types. The first member added to a group determines the type of machine the group can contain. By having multiple machines defined as members of a single group, you can perform a single operation on all machines by specifying the group as the target. NIM iterates through the list of machines in the group, and performs the operation on each member, in turn.

Group members can be excluded from NIM group operations by marking them for exclusion before performing the operation. Excluding a member marks the member list in the group representation, so NIM skips the member when it iterates through the list. Excluding a member does not change the definition of the client in the NIM database. For information on marking group members for inclusion and exclusion, see “Including and Excluding a Group Member from Operations on the Group” on page 40.

Machines can be added or removed from groups, as well as be members of multiple groups. When all members of a group are removed, the group definition in the NIM database is automatically deleted. For information on adding and removing group members, see “Adding New Members to Machine Groups” on page 38 and “Removing Members from Machine Groups” on page 39.

The command line syntax for defining a machine group is:

```
nim -o define -t mac_group -a
Attribute=Value ... \
MachineGroupName
```

where the following attributes are optional:

add_member=*Value* Specifies the name of a NIM client to add to the machine group. NIM automatically converts this attribute to a **member** attribute with an appropriate sequence number.

member=*Value* Specifies the name of a NIM client to add to the machine group. This attribute requires a sequence number.

Operations performed on machine groups are, by default, performed asynchronously on the non-excluded members of the group. NIM does not wait for an operation to complete on one group member before initiating the operation on the next member. When performing operations asynchronously, it is not possible for NIM to display all the output as it occurs on each client. Therefore, you should use the **lsnim** command to check the states of the group members to determine how far, and how successfully, the operations have executed. If errors do occur, the log files on client machines can be viewed using the NIM **showlog** operation.

To change the behavior of NIM group operations from asynchronous to synchronous, use the **async=no** attribute when running the **nim** command.

The number of machines permitted in a machine group is not explicitly limited by NIM. However, the following factors limit the number for practical reasons:

Operation being Performed Operations that are not resource-intensive (such as the **maint** or **showlog** operations) may be performed on a group containing any number of machines. Operations that are resource-intensive (such as **cust** or **bos_inst**) are limited by the throughput of the network, the disk access throughput of the installation servers, and the platform type of servers.

NFS Export Limitations The maximum number of hosts to which a file or directory may be exported with root permissions is limited by NFS to 256. Also, the length of a line in an exports file has an upper limit which could determine the maximum number of machines permitted in a group. See “Increasing the Number of Hosts to Which NIM May NFS-Export a Resource” on page 64 for details on how to increase the number of machines to which a resource may be allocated.

Resource Groups

Resource groups are used to represent collections of NIM resources. A resource group can contain multiple **installp_bundle** and **script** resources, and one resource from each of the other types. If a resource group is allocated or specified for use in a NIM operation, all applicable resources in the group are allocated to the target. The use of resource groups can save NIM administrators from having to repeatedly specify the same set of resources individually, when the allocation of a single resource group would suffice.

The command line syntax for defining a resource group is:

```
nim -o define -t res_group -a
default=Value \
-a ResourceType=ResourceName ... ResourceGroupName
```

where the following attributes are optional:

default=*Value* Specifies whether a resource group should be made the default. The default value is **default=no**.

ResourceType Specifies the type (for example, **spot**, **lpp_source**, **script**, etc.) and name of the resource to add to the group. One resource of each type may be specified, except for **script** and **installp_bundle** resources, which may have multiple resources participate in an operation.

The allocation of individual resource group members can be overridden by specifying additional resource attributes for the members to be changed.

For example, the resource group, res_grp1, contains the spot1, lpp_source1, bosinst_data1, script1, and resolv_conf1 resources. To use the resource group to perform an **rte bos_inst** operation on client1, but using no **bosinst_data** resource, and using resolv_conf2 instead of resolv_conf1, use the following command:

```
nim -o bos_inst -a source=rte -a
group=res_group1 \
-a bosinst_data= -a resolve_conf=resolv_conf2 client1
```

A resource group can be specified as the default set of resources to use for all NIM operations. This is done by setting the master's **default_res** attribute to the name of the resource group that will be the default. When a default resource group is defined, the applicable member resources will always be automatically allocated during NIM operations, unless they are specifically overridden.

To set the default resource group to res_group1, enter:

```
nim -o change -a default_res=res_group1
master
```

or enter:

```
nim -o change -a default=yes
res_group1
```

To stop using a default resource group, enter:

```
nim -o change -a default_res=
master
```

or enter:

```
nim -o change -a default=no
res_group1
```

Chapter 9. Network Installation Management Commands

This chapter contains reference information about the following Network Installation Mangement (NIM) commands:

- lsnim Command
- nim Command
- nimclient Command
- nimconfig Command
- nimdef Command
- niminit Command

Chapter 10. Error and Warning Messages

This chapter contains information about Network Installation Management (NIM) error and warning messages, with suggestions for resolving specific problems.

If an error condition is detected when a NIM command is executed, the command returns an error message. If a NIM command returns a warning message, this indicates that either a less severe problem was encountered by NIM, or a problem was encountered in a command called by NIM, and the severity of the problem cannot be readily determined by NIM. In the latter case, additional messages or output from the command often reveal the true nature of the problem.

All NIM error messages begin with `0042` and are followed by a three-digit error code. All NIM commands use this three-digit error code as their exit code when fatal errors are detected. Warnings are also identified by a three-digit code but do not affect exit codes and are preceded by the `warning` label.

Information about each message is organized in the following manner:

Message	Indicates the warning or error message ID number returned by the command.
Explanation	Describes what is likely to have caused the message to be displayed.
User Action	Suggests a possible resolution to the problem.

Notes:

1. If you require usage information for a NIM command, type the command without any arguments or with a question mark as an argument (for example, `nim -?`). Additional information can be obtained from the **lsnim** command, which provides several options to display NIM help and usage information. For more information, refer to the **-q**, **-O**, and **-P** options of the **lsnim** command. You can also use the **lsnim -p -a** command to display information for all NIM classes, subclasses, types, and attributes. For example, to determine the list of valid values for an attribute, enter:

   ```
   lsnim -p -a AttributeName
   ```

2. If a User Action for a given error or warning specifies using the **lsnim** command for recovery hints, and if you are operating from a NIM client, use **nimclient -l** *lsnimOperations*, substituting the suggested **lsnim** options as appropriate.
3. In some cases, a **nim** or **nimclient** operation that is being blocked because an object is in a particular state may be permitted with the use of the **force** option (the **-F** flag). However, by using the **force** option, you may adversely affect part of the NIM environment by forcing an operation that should only proceed after other actions are complete. Use error messages that are displayed without using the **force** option to determine if the **force** operation is a reasonable action.
4. If you believe that your problem is the result of a software defect, or if the User Actions provided here do not provide adequate resolution to a problem, contact your point of sale.

Message	0042-001
Explanation	An error was detected by an underlying NIM method (a subcommand). This message describes where the error occurred with respect to the NIM client or master and may be useful in troubleshooting the problem. The messages that are displayed subsequent to this error are normally the true source of the problem.
User Action	Read the additional information and error messages, and refer to their explanation and recovery hints as appropriate.

Message	0042-002
Explanation	An interrupt signal has been received, perhaps because the user entered Ctrl-C or used the **kill** command on a NIM process.

User Action The NIM operation that was active has been interrupted. Perform the operation again.

Note: This error is expected if it occurs after the **nimclient -o bos_inst** operation is performed on a client.

Message 0042-003 and 0042-004
Explanation An error has been returned from a system call.
User Action Fix the condition that caused the system call to fail and perform the operation again.

Message 0042-005
Explanation The Object Data Manager (ODM) has returned an error.
User Action Refer to the *Messages Guide and Reference* for specific details of the error. Fix the ODM problem and perform the NIM operation again.

Message 0042-006
Explanation Generic error message used for rarely occurring NIM errors.
User Action Phrases contained in this error message are constructed from debug information and from messages returned by commands called by NIM. If the content of the message does not give insight into the true cause of failure, contact your point of sale.

Message 0042-007
Explanation An internal NIM error has occurred.
User Action Try the operation again.

Message 0042-008
Explanation NIM has attempted to establish socket communications with a remote machine, and it has refused the connection.
User Action If the failing operation occurred on the master, verify that the master has **rsh** permissions on the client and that **inetd** is active on the client; otherwise, verify that the **nimesis** daemon is active on the master. If the failing operation was the **niminit** command on the client, a possible cause of failure is that the master does not have a network object that corresponds to the client's network. A network object that represents the client's network needs to be added to the database on the master; then a route needs to be added from the master's network to the client's network.

If the failure occurs during operations initiated from a client, using the **nimclient** command, or during a NIM installation of the base operating system, the **cpuid** attribute on the client's machine definition may be obsolete (for example, if the machine's system planar was recently replaced). To guarantee that this is not the case, erase the **cpuid** from the machine definition by issuing the following from the master:

```
nim -Fo change -a cpuid= ClientName
```

Message 0042-011
Explanation The **/etc/niminfo** file is not accessible.
User Action The **niminfo** file is required by all NIM commands and methods. This file is created when the **bos.sysmgt.nim.master** and **bos.sysmgt.nim.client** packages are configured. If this file is not available, this indicates that the NIM package has not been initialized or that this file has been deleted. To create the **niminfo** file, execute the **nimconfig** command on the master or the **niminit** command on the client. To recreate a deleted or corrupted **niminfo** file, enter from the master:

```
nimconfig -r
```

OR enter from the client:

```
niminit -aname=ClientName -amaster=MasterHostName -amaster_port=MasterPortValue
```

Message 0042-012
Explanation The specified command may only be executed on the master.
User Action Execute the desired operation on the NIM master.

Message 0042-013
Explanation The global lock used for synchronized access to the NIM database could not be obtained.
User Action Try the operation again. If the same error is returned, verify that there are no active NIM commands. If this is true, remove the **/var/adm/nim/glock** file and try the operation again. If the file does not exist and the error persists, contact your point of sale.

Message 0042-014
Explanation An internal NIM error has occurred.
User Action Perform the **remove** operation on the NIM object followed by the appropriate **define** operation.

Message 0042-015
Explanation A syntax error has been detected.
User Action Refer to the appropriate man page for the NIM command and try again using valid syntax.

Message 0042-016
Explanation An invalid option has been specified.
User Action Refer to the appropriate man page for the NIM command and try again using valid syntax.

Message 0042-017
Explanation An invalid value was specified for an option argument.
User Action Refer to the appropriate man page for the NIM command and try again using valid syntax.

Message 0042-018
Explanation A required option was not supplied.
User Action Refer to the appropriate man page for the NIM command and try again using valid syntax.

Message 0042-019
Explanation An option that requires an argument was specified without its argument.
User Action Refer to the appropriate man page for the NIM command and try again, specifying the missing argument.

Message 0042-20
Explanation An operand was required but not supplied. Usually, the operand is the NIM object to which a given operation is being applied (that is, a NIM name for a network, machine or resource object that is the target of the NIM operation).
User Action Refer to the appropriate man page for the NIM command and try again using valid syntax. If you do not know the name of an operand, and if the failing operation was targeted toward an existing NIM object, enter:

```
lsnim -l -t ObjectType
```

OR

```
lsnim -l
```

to determine the operand name.

Message 0042-021
Explanation A NIM attribute was required for the operation.

User Action Specify the missing attribute. If the failing command is the **nim** or **nimclient** command, to obtain a list of attributes, enter from the master:

```
lsnim -q ObjectName
```

OR

```
lsnim -q -t ObjectType
```

OR enter from the clients:

```
nimclient -l lsnimOptions
```

For the other NIM commands, see the appropriate NIM man page.

Message 0042-022
Explanation A value was specified that exceeds the bounds of acceptable values.
User Action Supply a value within the acceptable bounds.

Message 0042-023
Explanation The specified value is not valid.
User Action Try the command again with a valid value. To determine the valid values for classes of objects and operations as they pertain to those objects, enter:

```
lsnim -Pc ObjectClass
```

AND

```
lsnim -POc ObjectClass
```

where `ObjectClass` is one of machines, networks, or resources.

Message 0042-024
Explanation An invalid NIM object type was specified.
User Action Specify a valid NIM object type. See user actions for error 023 for **lsnim** options to determine a valid object type.

Message 0042-025
Explanation The specified operation cannot be supplied to the specified NIM object.
User Action Specify an operation that can be applied to the object. Enter `lsnim -O ObjectName` for a list of valid operations that can be applied to the object.

Message 0042-027
Explanation The specified object is missing an attribute that is required to complete the specified operation.
User Action Redefine the object that is missing an attribute by performing the **remove** operation followed by the **define** operation.

Message 0042-028 and 0042-029
Explanation The specified information cannot be supplied in the current context.
User Action Try the operation again without supplying the offending attribute.

Message 0042-030
Explanation A sequence number was opened to an attribute that doesn't allow sequence numbers.
User Action Try the operation again without a sequence number on the offending attribute.

Message 0042-031
Explanation An internal NIM error has occurred. NIM is unable to generate a unique object ID.
User Action Try the operation again.

Message 0042-032
Explanation The specified value for the attribute is not unique and it must be.
User Action Supply a unique value for the attribute.

Message 0042-033
Explanation The specified value is not unique and it must be. An attribute with a sequence number requires a unique value.
User Action Supply a unique value.

Message 0042-034
Explanation The specified value is not unique and it must be.
User Action Supply a unique value.

Message 0042-035
Explanation NIM was attempting to access an attribute that had the specified characteristics, but the attribute doesn't exist.
User Action Make sure the attribute exists and retry the operation.

Message 0042-036
Explanation The **define** operation failed for a resource because the specified server does not have a standalone configuration.
User Action Try the operation again using a NIM client that is a standalone machine.

Message 0042-037
Explanation The NIM state of the specified object prevents the operation from succeeding.
User Action NIM states are used to synchronize activity among NIM objects. To perform the desired operation, the state of the specified object must be changed. If the specified object is in an unexpected state, check the system to make sure another user or process is not manipulating the object. Use the **reset** operation to set the object to a known state and try the operation again.

Message 0042-038
Explanation An object that NIM would operate on is already locked and thus cannot be operated on.
User Action NIM object locks are used to synchronize activity among NIM objects. These locks are temporary, so try the operation again after some delay. The value of the lock is the process ID of a NIM process that is using the lock. If the lock persists and no NIM commands are active, reset all NIM locks by stopping the **nimesis** daemon, then restarting it.

Message 0042-039
Explanation The operating system version or release level of the specified object is unacceptable.
User Action Perform the desired operation on objects that have the appropriate operating system version and release levels.

Message 0042-040
Explanation A NIM object could not be removed because it is being used by some other NIM object.
User Action Remove all references to the object to be removed before the **remove** operation is specified. If NIM states are such that you cannot remove references to the object and you want to remove the object anyway, provide the **-F** flag to the **remove** operation.

Message 0042-041
Explanation A specified value has already been defined to NIM.
User Action Specify a value that isn't already known to NIM.

Note: If **/etc/niminfo** is the value and the NIM command producing this error is **niminit**, this means that **niminit** has already been performed. If you want to reinitialize your NIM master or client, deinstall the appropriate fileset, and then reinstall and reconfigure the NIM master or client fileset.

Message 0042-042
Explanation The specified machine could not be reached with the **ping** command from the master.

User Action If the operation you were attempting to perform requires that the target machine be running and that it can be reached, then verify that the machine is currently running. If not, turn it on; otherwise, perform network diagnostic procedures to determine why the master could not reach the target machine.

Message 0042-043
Explanation The remove operation cannot be performed, because the target machine currently serves a NIM resource that has been allocated for use. Performing the operation at this time could lead to processing failures on clients that are attempting to use the served resources.
User Action You need to deallocate all resources that the target serves before you can remove the machine.

Message 0042-044
Explanation You have specified a NIM attribute without an accompanying value. Most NIM attributes can only be specified with a value assigned to them in the form of *attr=value*.
User Action Retry the operation with a value assigned to the specified attribute.

Message 0042-045
Explanation Some NIM attributes can be added to an object's definition more than once. In these cases, a sequence number is used to uniquely identify each attribute of that type. In this case, you have specified an attribute of this type without its required sequence number and, therefore, NIM is unable to determine which attribute you are attempting to specify.
User Action Verify the sequence number and try the operation again.

Message 0042-046
Explanation NIM was unable to perform an operation on the specified file. This may be due to the permissions on the file. The file usually needs read, write, and, in some cases, execute permissions for root.
User Action Change the permissions of the specified file and try the operation again.

Message 0042-047
Explanation Some types of NIM resources may only be used by specific machine types. In this case, you attempted to allocate a NIM resource to a type of machine that is not allowed to use that type of resource.
User Action Specify a resource type that the machine is allowed to use when performing allocation for the target machine.

To determine the valid resource types, enter:

```
lsnim -p -s ResourceSubclassForMachineType
```

To view the subclasses that are available, enter: `lsnim -p -S`

Message 0042-048
Explanation When resource allocation is requested, NIM verifies that the designated client has the potential to communicate with the server of the resource. NIM does this by checking the NIM routing between the network that the client's primary interface connects to and all the networks that the server connects to. In this case, a NIM route is missing between the client and the server.
User Action Either establish a NIM route between the client and the server or choose a different resource to allocate.

Message 0042-049
Explanation Only one resource of this type may be allocated to the client and one has already been allocated.
User Action Choose the resource that you want to use and deallocate the currently allocated resource of this type if you want to use the new one.

Message 0042-051
Explanation NIM was unable to resolve a host name to an IP address or the other way around.

User Action All host names that are used in the NIM environment must be resolvable. Perform the appropriate network administration tasks to ensure that the specified host name is resolvable and try the operation again.

Message 0042-052

Explanation One or more NIM resources are still allocated to the machine that you have requested to be removed from the NIM environment. To remove a machine, it cannot have any resources allocated to it.

User Action Deallocate all resources that have been allocated to the target machine and try the operation again.

Message 0042-053

Explanation You have specified the name of a NIM object that does not currently exist in the NIM environment. NIM can only operate on objects that have been defined to NIM.

User Action Verify that you have spelled the name of the object correctly and that it has already been defined. The name of a target machine for a NIM operation must be the NIM name, not the host name. Enter:

```
lsnim -l -t ObjectType
```

OR

```
lsnim -l
```

to obtain listings of currently defined objects in the NIM environment. If you need to define the object, use the **define** operation.

Message 0042-055

Explanation Many NIM operations require a source for installable images. You have specified a source that cannot be used for this operation. Examples of valid sources for NIM operations are:

- **/dev/rmt0**, **/dev/cd1** for **lpp_source** definition
- **rte**, **spot**, **mksysb** for **bos_inst** operation

User Action Try the operation again using a source that the operation can use.

Message 0042-056

Explanation You have specified the same attribute assignment more than once.

User Action Try the operation again using only one instance of the attribute assignment.

Message 0042-058

Explanation You have attempted to allocate a **SPOT** to a client whose primary network interface type or platform is not supported by the **SPOT**. For a client to use a **SPOT**, the **SPOT** must support the network interface type and platform of the client's primary interface.

User Action Install the appropriate device support into the **SPOT**, which will allow the **SPOT** to support the client's primary interface type and platform, or choose a different **SPOT** that supports the client's primary interface type and platform.

Message 0042-059

Explanation In an attribute assignment (in the form of **attr**=**value**), the **value** you have specified represents a NIM object whose type conflicts with the object type of the specified **attr**.

User Action Try the operation again using the **attr** that corresponds to the type of object that **value** represents.

Message 0042-060

Explanation You have specified multiple attribute assignments for an attribute that may only be specified once.

User Action Try the operation again, using only one instance of the attribute.

Message 0042-061

Explanation You have requested an operation to be performed on a NIM resource object that is currently allocated for client use. NIM is not allowing this operation to be performed because it may interrupt the client's use of the resource.

User Action Try the operation again when the resource is not allocated for client use. If necessary, try the **force** option (**-F** flag) to disregard the preventive check by NIM. In some cases, NIM will allow the operation to be performed.

Message 0042-062

Explanation The NIM object that was operated on is missing something that is required for its definition to be complete.

User Action List information about the object using the **lsnim** command. Each item that is missing from the object's definition will be represented by a missing attribute. Perform the appropriate NIM operation that will add the missing item to the object's definition. For a **SPOT**, if network boot images are missing, apply the **check** operation to the **SPOT**. If software filesets are missing from a **SPOT**, allocate an **lpp_source** that contains the required filesets and apply the **cust** operation to the **SPOT**.

Message 0042-063

Explanation Some NIM operations require access to one or more NIM resources to complete successfully. This access is granted through the **allocate** operation. In this case, you have not allocated all the resources that are required for this operation.

User Action Allocate all the required resources and try the operation again. For a list of required and optional resources for a given operation, enter:

```
lsnim -q Operation ObjectName
```

OR

```
lsnim -q Operation -t ObjectType
```

Message 0042-064

Explanation The machine that is the target of the requested operation currently serves a NIM resource that is allocated for client use. The requested operation cannot be performed until all resources that the target serves have been deallocated for use.

User Action Deallocate all resources that the target serves and try the operation again.

Message 0042-065

Explanation You have specified a name that is reserved for NIM internal use only.

User Action Try the operation again using a different name. To determine what names are reserved, enter:

```
lsnim -a reserved
```

Message 0042-066

Explanation You have specified one or more characters that are not allowed in NIM object names. NIM uses regular expressions to perform many of its operations, so any character that has special meaning for regular expressions cannot be used (for example, ^). Also, any character that has special meaning to the shell cannot be used (for example, /).

User Action Try the operation again using valid characters.

Message 0042-067

Explanation You have requested an operation to be performed on a NIM object that has been reserved for NIM internal use only.

User Action Try the operation again, using a NIM object that is not reserved. To determine what objects are reserved, enter:

```
lsnim -a reserved
```

Message 0042-069

Explanation The requested operation cannot be performed at this time because it conflicts with the current NIM state of the target. NIM uses states to synchronize NIM activity so that operations don't interfere with each other.

User Action Try the operation again when the state changes or, if necessary, try using the **force** option (**-F** flag). In some cases, NIM will allow you to override this state checking.

If you encounter this error as a result of trying to remove, using the **reset** operation, the **boot** resource from a client that incorrectly has a state of ″ready for a NIM operation″, you can remove the **boot** resource from the NIM master by entering:

```
/usr/lpp/bos.sysmgt/nim/methods/m_dealloc_boot client_name
```

where `client_name` is the name of the NIM object for the client.

Message 0042-073

Explanation To perform customization on a machine, NIM constructs a shell script that is executed on the target. To construct this script, some type of resource that can be used for customization must be used. In this case, NIM could not create the customization script because no resources have been allocated to the target that could be used for customization purposes.

User Action Allocate one or more resources that can be used for customization and try the operation again. To display the subclass of resources that can be used for customization, enter:

```
lsnim -p -s cust_res
```

Message 0042-074

Explanation You have specified an attribute assignment in which the **value** represents a relative path name. NIM only allows absolute path names (that is, path names that begin with /) to be used.

User Action Try the operation again, using an absolute path name.

Message 0042-075

Explanation The requested operation requires that a NIM resource be exported for a machine's use. In this case, NIM attempted to export the resource but an error was returned by an NFS utility.

User Action Fix the error condition that the NFS utility reported and try the operation again.

Message 0042-076

Explanation You have specified a port number that is already in use.

User Action Try the operation again, using a port number that is currently not being used. Check the **/etc/services** file.

Note: NIM uses both the specified port number and its successor. Therefore, ensure that the port number after the specified port number is also free.

Message 0042-077

Explanation The **niminit** command is used to join the NIM environment. When executed, this command attempts to add routing information that the NIM master has determined the client needs to participate in the NIM environment. In this case, one or more of the required routes could not be added.

User Action Perform the appropriate network diagnostic task to determine why the route could not be added.

Message 0042-078

Explanation You have specified a change to a NIM routing attribute in which the destination network is different from its current value. This is not allowed because only the gateway field of the routing attribute may be changed.

User Action If you are trying to change the connectivity between NIM networks, then you must remove the current NIM route by supplying a NULL value for the appropriate routing attribute. Otherwise, specify the same destination network when attempting to change the gateway field of the routing attribute.

Message 0042-079
Explanation In the NIM environment, one resource may depend on another for information. In this case, an allocated resource has a dependency on the resource you have specified for deallocation.
User Action Deallocate the resource that is dependent on the resource causing the error.

Message 0042-081
Explanation NIM uses NFS to make remote resources available for client use. To avoid NFS export errors, NIM enforces some restrictions on where a resource can be defined. In general, a NIM resource cannot be defined within a directory that is already a NIM resource. Conversely, a NIM resource cannot be defined for a directory that already contains an existing NIM resource.
User Action Move the resource to a location that adheres to NIM export rules and try the operation again.

Message 0042-083
Explanation Each network communications adapter has an associated network hardware address that is unique. In this case, you attempted to define a NIM network interface using a network hardware address already being used by a NIM machine object.
User Action Only one NIM interface attribute may be defined for each network communications adapter a client might have. If you are attempting to add another interface definition, then verify that the hardware address is correct. If so, then you must first change the interface attribute that is currently using that address. If not, try the operation again with the correct hardware address.

Message 0042-084
Explanation The machine has already been configured to be a NIM master.
User Action If you want to reconfigure the machine as a NIM master, enter `nim -o unconfig master`, then deinstall and reinstall the master fileset. You may then run the **nimconfig** command.

Message 0042-086
Explanation You have attempted to add a NIM route between two NIM networks that already have a NIM route between them. Only one NIM route can be specified between any two NIM networks.
User Action If you are attempting to change NIM routing, delete the existing NIM route and try the operation again.

Message 0042-093
Explanation NIM attempted to create a directory, and the **mkdir** command returned an error.
User Action Fix the error reported by the **mkdir** command and try the operation again.

Message 0042-109
Explanation To complete the requested operation, NIM requires information about one or more file systems about which it was unable to obtain information.
User Action Verify that the file systems exist. If not, either specify the correct path name when performing the NIM operation or redefine the NIM environment so that all the participating file systems exist.

Message 0042-111
Explanation When a sequence number is specified for a NIM attribute, it must be within a specific range. You have specified a sequence number that falls outside of the acceptable bounds.
User Action Try the operation again using a sequence number that is within the acceptable bounds.

Message 0042-113
Explanation To complete the requested operation, NIM requires information about the size of one or more objects, which NIM was unable to determine.
User Action If the object is a file or directory that does not exist, then create the file or directory and try the operation again.

Message 0042-118

Explanation You have requested to change characteristics of a NIM network on which there is currently one or more active NIM operations. NIM is not allowing the change because changing the network characteristics at this time could result in failures in the active operations.

User Action Wait until the NIM resources allocated to machines that use the network being changed have been deallocated and try the operation again.

Message 0042-121

Explanation An invalid value has been specified.

User Action Try the operation again, using a valid value.

Message 0042-124

Explanation An NFS option was specified that is not supported.

User Action Try the operation again using valid NFS options. Refer to NFS Problem Determination in *AIX 5L Version 5.1 System Management Guide: Communications and Networks*.

Message 0042-129

Explanation An invalid resource type was specified for this operation.

User Action Use the **lsnim -q** *Operation* **-t** *TargetType* command to view a list of required and optional resources for *Operation* when applied to *TargetType*.

Message 0042-130

Explanation You have specified an attribute that cannot be used for this requested operation.

User Action Try the operation again, without using the attribute you specified.

Message 0042-131

Explanation You have specified two or more attributes that conflict with each other.

User Action Try the operation again, using only one of the attributes.

Message 0042-132

Explanation You have specified a **value** for an attribute assignment that is invalid in the context in which the operation is being performed.

User Action Try the operation again, using a different **value** for the attribute assignment.

Message 0042-133

Explanation The physical entity that is represented by the NIM resource object you have requested to be deleted could not be deleted.

User Action Delete the file or directory, using the **rm** command.

Message 0042-134

Explanation The operation you requested requires the designated target to reboot using a network boot image. NIM has automatically initialized the environment to enable the target to do this; however, NIM was unable to force the target to load the network boot image.

User Action Intervention at the target is required. Follow the procedures for initiating a BOOTP request.

Message 0042-135

Explanation To synchronize NIM operations that can be initiated from a client or on the master, NIM keeps track of which machine (the client or the master) is used to allocate the first resource to the client; this machine is said to be in control. For example, if the first resource allocation occurs from the client, then the client is in control. Once a machine has control, it remains in control until all resources for that client have been deallocated. You have requested an operation to be performed from a machine that is currently not in control of the specified target.

User Action Perform the desired operation from the machine that is in control of the target, or from the controlling machine deallocate the resources (to remove the control), or override this behavior by using the **force** (**-F** flag) option when performing the operation from the master.

Message 0042-136

Explanation The requested operation cannot be performed because a NIM route does not exist between two machines that participate in this operation.

User Action Establish a NIM route between the networks.

Message 0042-137

Explanation The **/etc/niminfo** file contains information about the NIM environment that all NIM commands require. In this case, the **/etc/niminfo** file is missing some information that is required to continue, which indicates that the file has been corrupted.

User Action Reinitialize the NIM environment.

Message 0042-138

Explanation Unable to update the **rhost** file.

User Action Edit the **$HOME/.rhosts** file for root on the client machine to add an entry for the host name of the NIM master.

Message 0042-139

Explanation The process of installing a machine prevents any attached disks from being used as the source for installable images. You have allocated a resource to the target of the install operation that is served by the target itself.

User Action Deallocate the resource, allocate another resource of this type that is served by another machine, and try the operation again.

Message 0042-140

Explanation You have requested that a machine object be removed from the NIM environment and this has been done; however, NIM was unable to remove the **/etc/niminfo** file on the machine that has been removed from the NIM environment.

User Action Remove the **/etc/niminfo** file from the machine that was removed from the NIM environment.

Note: Verify that the **.rhost** permissions for the master have been removed from the client.

Message 0042-141

Explanation By specifying an attribute assignment with a NULL value, you have requested NIM to remove the specified *attr*. However, in this case, the specified *attr* is not currently part of the specified object's definition.

User Action Try the operation again, using an attribute that is part of the object's definition.

Message 0042-142

Explanation All attribute values must be unique. You have specified a *value* in an attribute assignment that already exists.

User Action Try the operation again, using a unique *value* for the attribute.

Message 0042-143

Explanation Some NIM attributes can only be added to an object's definition once. In this case, you have specified an attribute of this type when one already exists for the specified object.

User Action Only one attribute of this type can be used in the object's definition. Perform the change operation on the object if you want to replace the current value with a new one.

Message 0042-144

Explanation Some NIM attributes require a unique sequence number so that NIM can distinguish between multiple attributes of that type. In this case, you have specified a sequence number that is already being used.

User Action Try the operation again, using a sequence number that is not currently being used. To display the sequence number that are being used, enter:

```
lsnim -a AttributeName ObjectName
```

Message 0042-145

Explanation You have specified an attribute that does not exist in the NIM environment.

User Action Try the operation again, using a valid NIM attribute. To display a list of valid attribute names, enter:

```
lsnim -p -s info_subclass
```

Message 0042-146

Explanation You have specified an object type that does not exist in the NIM environment.

User Action Try the operation again, using a valid NIM object type. On the NIM master, the **lsnim** command can be used to display the valid NIM object types.

Message 0042-147

Explanation You have attempted to execute a NIM command on the NIM master that can only be executed on NIM clients.

User Action Execute the command on a NIM client.

Message 0042-148

Explanation The information contained in the specified attribute is no longer valid.

User Action Change the information in the attribute to reflect valid information and try the operation again.

Message 0042-150

Explanation Any directory used to store NIM resources must be local to the machine that serves those resources. This is required because NIM can only NFS export local directories. In this case, you have specified a directory that is not local to the designated server of the directory. NIM has obtained this information from the file system of the designated server and the **vfstype** listed corresponds to values in the **/usr/include/sys/vmount.h** file.

User Action Either copy the desired resources onto the designated server and perform the operation again, or specify the correct server when performing the operation.

Message 0042-151

Explanation For NIM to use a file, it must be of a specific type. In this case, you have specified a file whose type cannot be used by NIM. NIM has obtained this information from the file system of the designated server of the file and the file type corresponds to values in the **/usr/include/sys/mode.h** file.

User Action Change the file type of the file and try the operation again.

Message 0042-152

Explanation When an **installp** operation is performed on a **SPOT**, the root directories of all diskless and dataless clients that use that **SPOT** must be synchronized with the changes made within the **SPOT**. In this case, one or more errors occurred when performing the **root sync** operation on a root directory.

User Action Investigate why some of the root syncs failed and perform the operation again. The **nim.installp** log for the client root is located in *RootResrcParentDir*/*ClientName*/**var/adm/ras**.

Message 0042-153

Explanation For NIM to use a file, it must have specific file permissions. In this case, you have specified a file whose permissions conflict with those required by NIM. NIM has obtained this information from the file system of the designated server of the file, and the value of the file permissions comes from the **/usr/include/sys/mode.h** file.

User Action Change the file permissions of the file and try the operation again.

Message 0042-154

Explanation For NIM to use a file, it must exist. You have specified a file that does not exist.

User Action Create the file and try the operation again.

Message 0042-155

Explanation For NIM to keep diskless and dataless root directories in sync with their corresponding **SPOTs**, NIM requires that the client's root directory be served from the same machine as its **SPOT**. In this case, you have requested a resource to be allocated that violates that requirement.

User Action Try the operation again using resources that do not violate the NIM requirement.

Message 0042-156

Explanation You have requested an operation to be performed that involves a directory that does not exist.

User Action Create the missing directory and try the operation again.

Message 0042-157

Explanation The operation you have requested could not be performed because a required file could not be accessed.

User Action Create the missing file and try the operation again. For example:

- If the missing file is a boot image with a name whose format is *SpotName.NetworkInterface.Platform* (for example, `myspot.tok.up`), recreate the boot image by performing the check operation on the **SPOT**.
- If the missing files are directories with which **root** or **paging** resources are associated, delete the resource definition with the **remove** operation, create the directories, and then redefine the resource.
- If a **SPOT**'s **image.template** file is missing, this indicates that the **SPOT** has been corrupted or was not constructed successfully. To recover, you may need to remove and rebuild the **SPOT** with the **remove** and **define** operations.

Message 0042-158

Explanation The operation you have requested requires NIM to modify a file that it was unable to modify successfully.

User Action Check the file permissions on the file and try the operation again.

Message 0042-159

Explanation Required software is missing which prevents the target machine from acting as a **SPOT** server.

User Action Install the missing software and retry the operation.

Message 0042-160

Explanation The operation you requested requires the construction of network boot images and NIM was unable to do that.

User Action Fix the problem that prevented the network boot images from being constructed and try the operation again.

Message 0042-161

Explanation There is insufficient free disk space to complete the requested operation.

User Action Increase the amount of available space, as detailed in the error message.

Message 0042-162

Explanation To perform the requested operation, NIM requires an **lpp_source** type resource object that has the **simages** attribute as part of its definition. This attribute is used to designate that an **lpp_source** contains the total set of optional packages that are required to support NIM install operations. In this case, you have not supplied an **lpp_source** that fulfills this requirement.

User Action Try the operation again using an **lpp_source** that has the **simages** attribute in its definition.

Message 0042-163

Explanation NIM coordinates access between a client and the server of the resource. To do this, NIM must identify a network interface that can be used by the client. This becomes a complex problem when the server has more than one network interface. NIM uses a connectivity algorithm to establish which network interface to use. This error message occurred because the connectivity algorithm detected a problem with the client's routing and the interface the algorithm has selected to use. NIM does not allow the interface on the server that the client uses as a gateway to be used to serve resources because the operation requiring the resource could fail.

User Action If the server has other network interfaces that are not known to NIM, change the server machine object to add the interfaces.

Define a NIM route between the client's primary network and one of the other networks to which the server connects.

Message 0042-164
Explanation Some NIM operations do not allow the source of installable images to be a CD-ROM. NIM is not always able to construct an environment that supports the use of a CD-ROM for the operation being performed. This is true for the operation you tried to perform.
User Action Try the operation again using a different source for installable images.

Message 0042-165
Explanation Some attributes can only be specified together; others are mutually exclusive. In this case, you specified one or more attributes that conflict.
User Action Try the operation again, omitting the attribute that was in conflict. For example, the **ring_speed** and **cable_type** attributes cannot be used with the same **if** attribute; the one you should use depends on the type of network interface referenced by the corresponding **if** attribute.

Message 0042-166
Explanation The **if** attribute specifies network interface information, which includes a reference to the network object that the interface connects to. In this case, you have omitted a required attribute which is associated with the **if** attribute.
User Action Try the operation again, including the required attribute. For example, the **ring_speed** attribute corresponds with the Token-Ring network interface, and the **cable_type** attribute corresponds with the Ethernet network interface.

Message 0042-167
Explanation The device which you have specified as the source for the IPL ROM emulation, does not contain a valid, bootable image of the IPL ROM emulation.
User Action If the specified device has media in it, this media either does not contain the IPL ROM emulation, or the media has been corrupted. Remake the IPL ROM emulation, and try the operation again. If the specified device has no media in it, make the IPL ROM emulation, put it in the device, and try the operation again.

Message 0042-168
Explanation You have specified that the originating and destination network are the same. Machines that are on the same network do not need routing to communicate; therefore, adding a route from a network to itself is not allowed.
User Action Specify a different originating and destination network when adding a NIM route.

Message 0042-169
Explanation You have allocated an **lpp_source**, but you have not specified which filesets are to be installed using that **lpp_source**.
User Action Specify the filesets to install using the **filesets** attribute in the command, or by allocating an **installp_bundle** that contains a list of the filesets to install.

Message 0042-170
Explanation You entered a platform type that is not known to NIM.
User Action The valid platform types are **rs6k**, **rs6ksmp,** and **rspc**. Correct the platform type attribute and try the operation again.

Message 0042-171
Explanation Not all platform types are supported on all configuration types. For example, the diskless configuration type is not supported on the platform type **rs6ksmp**.
User Action Use the correct platform type and try the operation again.

Message 0042-172

Explanation You have specified the incorrect name of the machine object for the NIM client machine. When the **niminit** command is used to rebuild the**niminfo** file, the master registration process checks the CPU ID of the machine with the value stored in the NIM database for the named machine. If the stored value does not match the value passed by **niminit**, this message is issued.

User Action Use the correct name and try the command again.

Message 0042-173

Explanation You specified that the **installp** command should expand file systems (using the **-X** flag) while specifying that NIM should not auto expand (using the **auto_expand** attribute). This is not an allowable combination for the command invoked.

User Action Use either the **-X** flag or the **auto_expand** attribute, but not both.

Message 0042-174

Explanation You specified an invalid value for an attribute whose only valid values are **yes** and **no**.

User Action Retry the operation with a value of **yes** or **no** for the attribute indicated.

Message 0042-175

Explanation An unexpected result has been returned from a command that NIM tried to execute.

User Action Fix the problem that caused the executed command to fail and try the operation again.

If the command failed due to a shortage of space, its error messages indicating this should be displayed. Expand the indicated file system, (for most NIM operations use the **auto_expand** attribute) and retry the operation. If a space failure occurred during **SPOT** creation, and if the **bosboot** command failed to make boot images as a result, increase the free space and run the **check** operation.

If the command listed by NIM in this message is the **installp** command, check the **nim.installp** log for failure and recovery information. (For standalone client operations, this is located in the **/var/adm/ras** directory of the client. For **SPOT cust** and **maint**operations, this is located in *SPOTParentDir*/SPOTName**/usr/lpp/bos/inst_root/var/adm/ras** on the **SPOT**. For diskless and dataless clients, this is located in *RootResrcParentDir*/*ClientName***/var/adm/ras**.

Message 0042-176

Explanation The resource cannot serve as a support image (**simages**)**lpp_source**. When an **lpp_source** serves as a support image resource, it contains a minimal set of software packages for facilitating installation and the use of the base operating system.

User Action No action is necessary if this resource does not need to serve as a support images **lpp_source**. If the resource needs to be a support images **lpp_source**, add the missing software to the **lpp_source**. If the **lpp_source** is a directory, you can do this by copying the missing packages to the location of the **lpp_source** and running the **check** operation.

Message 0042-177

Explanation The operation you requested could not be completed due to insufficient free space in one or more file systems.

User Action Make more space available if possible, by extending the file system displayed. For most NIM operations, the **auto_expand** attribute is available to automatically expand file systems.

Message 0042-178

Explanation The **if** attribute is made up of four fields. The fourth field is optional in most cases. In this case. the network object that you specified (in field *one*) has more than one type of network. In this case, NIM requires that the fourth field has a value that specifies the logical device name of the network adapter.

User Action Add the appropriate value to the **if** attribute, and try the operation again.

Message 0042-179

Explanation You are attempting to remove an **if** or **other_net_type** attribute on which one or more NIM clients have a dependency.

User Action If this is not a mistake, remove the NIM clients that are dependent on the network, or remove the **if** attribute from the NIM client object definition.

Message 0042-180
Explanation The address of the machine that is being defined is not connected to the network that is represented by the specified network object.
User Action Define a network object that represents the physical network to which the machine is connected. Use this network object when defining the machine.

Message 0042-181
Explanation The **fix_query_flags** attribute has an illegal value. Use **lsnim -Pa fix_query_flags** for a list of legal values.
User Action Determine the correct flags and retry the operation.

Message 0042-182
Explanation A resource of one type cannot be allocated for the current operation at the same time as a resource of another type. Allocate one or the other, but not both.
User Action The resources specified are mutually exclusive. Determine which one is needed for the operation, and omit the other.

Message 0042-183
Explanation An attribute cannot be specified for the current operation when a type of resource is allocated. Use one or the other, but not both.
User Action The attribute and the resource specified are mutually exclusive. Determine which one is needed for the operation, and omit the other.

Message 0042-184
Explanation The network address (**net_addr**) or subnet mask (**snm**) cannot be changed for the network, because NIM clients are currently defined as being connected to that network. Remove the client definitions before changing the network.
User Action The **nimdef** command can be used to quickly redefine NIM clients after they have been removed to update the network definition.

Message 0042-185
Explanation Failed to link or copy files. Check permissions and file system space.
User Action Verify that space and inodes are available for the files and links specified in the error message.

Message 0042-186
Explanation Failed to copy setup programs. Either start NFS on the client or free 1000 512-byte blocks in the file system.
User Action Programs required to set up the operation could not be copied to the client system. Either start NFS on the client, or increase space in the file system specified in the error message.

Message 0042-187
Explanation Failed to expand file system.
User Action Attempt to manually expand the file system specified in the error message, then retry the operation.

Message 0042-188
Explanation Failed to NFS mount.
User Action Verify that NFS is running on both the resource server and the client specified in the error message. Retry the operation when the NFS problems have been resolved.

Message 0042-189
Explanation Failed saving existing boot image. Check space in the file system.
User Action Increase space in the file system specified by the error message, and retry the operation.

Message 0042-190
Explanation The key is *not* in the NORMAL position. Unattended installation cannot complete unless the key is in the NORMAL position.
User Action Turn the key on the client machine to the NORMAL position and retry the operation.

Message 0042-191
Explanation Unable to write the IPLROM emulation.
User Action The **mkboot** command failed to write the IPLROM emulation on the client. Boot the client manually over the network to begin the BOS installation.

Message 0042-192
Explanation Unable to find boot logical volume.
User Action Verify that a boot logical volume is defined for the machine. NIM attempts to use the **lslv -l hd5** command to determine the boot logical volume.

Message 0042-193
Explanation The client does not have an **.rhosts** entry for the master, or the client host ID is not resolvable.
User Action Verify that the client host name is resolvable by the master. Then verify that an entry exists for the master in the **$HOME/.rhosts** file for root on the client machine.

Message 0042-194
Explanation The client does not allow NIM **push** operations. Remove **/etc/nimstop** on %s if **push** operation is necessary.
User Action On the client machine, run the **nimclient -p** command to re-enable master push permissions.

Message 0042-195
Explanation Unable to order boot device list.
User Action An error was returned by the **bootlist** command on the client. If a network boot must be performed for a **bos_inst**, **diag**, or **maint_boot** operation, manually set the boot list and reboot the client, or follow the normal procedure to boot the client over the network.

Message 0042-196
Explanation The **set_bootlist** attribute is only valid when used in combination with the **no_client_boot** or **boot_client** attributes.
User Action Only specify the **set_bootlist** attribute to the **nim** command when changing the default behavior with the **no_client_boot** or **boot_client** attributes.

Message 0042-197
Explanation If the target machine has more than one interface for a given network type, the network adapter's logical device name must be specified in the **if1** attribute of the target machine's NIM definition when using the **force_push** attribute.
User Action Modify the client's **if1** attribute using the NIM **change** operation. Change the **if1** attribute to include one of the client's network adapter logical device names listed in the error message.

Message 0042-198
Explanation When converting a machine's **/usr** file system to a **SPOT**, the bos image on the media (**lpp_source**) being used to create the **SPOT** must match the bos image that was used to install the machine.
User Action When defining the **/usr SPOT**, use the same installation media that was used to install the machine originally. For example, if a machine was originally installed with AIX 4.1.3 and then updates were applied to bring the machine to AIX 4.1.4, the installation media that should be used when defining the **/usr SPOT** on the machine would still need to be the AIX 4.1.3 product media.

Message 0042-199
Explanation The **no_client_boot** and **boot_client** attributes may not be specified together.

User Action To avoid the possibility of giving conflicting instructions to the NIM command, do not supply both the **no_client_boot** and **boot_client** attributes in the same NIM operation.

Message 0042-204
Explanation The **mk_image** and **source** attributes are only valid when specified together.
User Action When creating a **mksysb** resource from a running client machine, use the **mk_image=yes** attribute to indicate that a **mksysb** should be created, and use the **source=***ClientName* attribute to specify the name of the client that is to be backed up.

Message 0042-205
Explanation The **bos.sysmgt.sysbr** fileset must be installed on the client to perform the system backup. You may install this fileset with the NIM **cust** operation.
User Action Install the **bos.sysmgt.sysbr** fileset on the client machine before retrying the operation.

Message 0042-206
Explanation There is already a resource allocated.
User Action Only one resource of the type specified can be allocated to the client. Deallocate the first resource before attempting to allocate the other.

Message 0042-207
Explanation Unable to allocate a resource to a client.
User Action Look for other NIM error messages that may accompany this error and which may provide more information about the problem. Verify that the resource specified is NFS-exportable to the client.

Message 0042-208
Explanation Unable to lock a client. This could mean that the client is already locked, or the name given does not refer to a valid NIM client.
User Action If another NIM operation is being performed on the same client, wait for the process to complete before retrying the operation. If no other NIM operations are being performed, stop and restart the **nimesis** daemon to remove locks.

Message 0042-209
Explanation The **mksysb_flags** attribute contains an illegal value. Use the **lsnim -Pa mksysb_flags** command to get a list of legal values.
User Action Specify the correct values for the **mksysb_flags** attribute, and retry the operation.

Message 0042-210
Explanation The maximum space required for the backup is greater than the amount of free space in the target file system. To ignore space requirements, use the **-F** flag when defining the **mksysb** resource.
User Action Either increase the space of the target file system where the **mksysb** is to be created, or use the **-F** flag as specified in the error message.

Message 0042-211
Explanation The member already exists in group.
User Action No additional action is required, since the member is already added to the group.

Message 0042-212
Explanation The member was not added to the group, because it is not a valid NIM name.
User Action The name of a member to add to a group was invalid. Verify that the member was specified correctly.

Message 0042-213
Explanation The group was not created, because it did not contain any valid members.
User Action A group must contain at least one member. Redefine the group with valid members to add it to the NIM environment.

Message 0042-214
Explanation Unable to add a member to a group.
User Action Look for other NIM error messages that may accompany this error and which may provide more information about the problem.

Message 0042-215
Explanation An invalid log type for the **showlog** operation was specified.
User Action Specify one of the valid log types listed in the error message.

Message 0042-216
Explanation An invalid log type for the **showlog** operation was specified for a **SPOT**.
User Action Specify one of the valid log types listed in the error message.

Message 0042-217
Explanation An invalid log type for the **showlog** operation was specified for a diskless or dataless machine.
User Action Specify one of the valid log types listed in the error message.

Message 0042-218
Explanation The log file is either empty or does not exist.
User Action No information is available in the log file for the machine or **SPOT** specified.

Message 0042-219
Explanation The object is incompatible with the group.
User Action The object cannot be added to the group, because its type is not allowed in the group. Machine groups can only contain one type of NIM client, and that type is determined by the first member added. Resource groups can only contain members whose types are resources.

Message 0042-220
Explanation You cannot have more than one resource of the specified type in a resource group.
User Action You must remove the current member with the specified type from the resource group before the new member with the same type can be added.

Message 0042-221
Explanation The group %s is being removed, because its single remaining member was removed during this operation.
User Action A group cannot be empty. Redefine the group with at least one member if it should remain in the NIM environment.

Message 0042-222
Explanation An unknown error occurred allocating resources to the machine.
User Action Look for other NIM error messages that may accompany this error and which may provide more information about the problem. Verify that the resource specified is NFS-exportable to the client.

Message 0042-223
Explanation Invalid input file. The file either cannot be read, is empty, or contains no valid entries.
User Action Verify that the file specified in the error message is the correct file for the operation.

Message 0042-224
Explanation The limit on the length of a line in an NFS exports file was exceeded. The **export** operation cannot be performed.
User Action Manually edit the **/etc/exports** and **/etc/xtab files** to remove any obsolete entries. The number of hosts to which NIM can NFS-export a resource can also be increased by setting the **restrict_nfs_exports** attribute to **no** on the master by running the **nim -o change -a restrict_nfs_exports=no master** command.

Message 0042-225

Explanation An error occurred while updating the exports file. Check for corruption in the file.

User Action Manually edit the **/etc/exports** and **/etc/xtab** files to fix any file corruption problems. Attempt to determine why NIM was unable to successfully update the files. Check file and directory permissions, and verify that file systems are not full.

Message 0042-226

Explanation A timeout occurred while attempting to initiate the operation on the client. The operation may not have started successfully.

User Action If the operation that was performed was **bos_inst**, the client only needs to be rebooted manually over the network to begin the installation. For all other operations, the problem is most likely due to network communication problems between the master and the client. Verify that the client is reachable by the master and that **rsh** permission is still granted by the client to the master.

Message 0042-227

Explanation The state of the machine indicates that it may not be ready for certain NIM operations.

User Action Check to see if any NIM operations are still being performed on the machine. If not, reset the state of the machine with the **nim -Fo reset** *MachineName* command. This returns the machine to the **ready** state so NIM operations can be performed on it. The **reset** operation does not deallocate resources, so deallocate resources if necessary using the **nim deallocate** operation.

Message 0042-228

Explanation Invalid release level.

User Action The release level of the resource is incomplete, or incorrectly specified. The level of the resource can be obtained by running the **lsnim -l** *ResourceName* command and viewing the **version**, **release**, and **mod** attributes. To correct the problem, either recreate the resource, or modify the NIM database to contain the correct level using the command on the NIM master:**/usr/lpp/bos.sysmgt/nim/methods/m_chattr -a** *Attribute*=*Value ResourceName*, where *Attribute* is **version**, **release**, or **mod**; *Value* is the correct value; and *ResourceName* is the name of the resource with the incorrect level specification.

Message 0042-229

Explanation When installing a system using a **mksysb** as the source for the installation, the level of the **SPOT** used for the installation must match the level of the **mksysb** image being installed. The release levels of the **SPOT** and the **mksysb** do not match.

User Action Create a **SPOT** that matches the level of the **mksysb** being installed, and use that **SPOT** when performing a **mksysb** BOS installation. The level of **mksysb** and **SPOT** resources can be obtained by running the **lsnim -l** *ResourceName* command and viewing the **version**, **release**, and **mod** attributes.

Message 0042-230

Explanation When installing a system using a **mksysb** as the source for the installation, the level of the **SPOT** used for the installation should match the level of the **mksysb** image being installed. If this convention is not followed, the installation may not complete successfully.

User Action Create a **SPOT** that matches the level of the **mksysb** being installed, and use that **SPOT** when performing a **mksysb** BOS installation. The level of **mksysb** and **SPOT** resources can be obtained by running the **lsnim -l** *ResourceName* command and viewing the **version**, **release**, and **mod** attributes.

Message 0042-231

Explanation A temporary list of software that should be installed is created and used for this operation. The list could not be created.

User Action Check previous error messages to understand why the error occurred. Correct the problem and try the operation again.

Message 0042-232

Explanation A temporary **installp_bundle** resource is created and used for this operation. The temporary resource could not be created.

User Action Check previous error messages to understand why the creation of the resource failed. Correct the problem and try the operation again.

Message 0042-233

Explanation The operation cannot be performed because the NIM Master is already initialized.

User Action Unconfigure the NIM Master and try the operation again.

Message 0042-234

Explanation You cannot restore a NIM database backup onto a machine that has an earlier level of the NIM master fileset installed. For example, a NIM database backup of a system with level 4.2.0.0 of the NIM master cannot be restored to a system that has a level of the NIM master lower than 4.2.0.0.

User Action Install a level of the NIM master fileset that is at the same level or a later level than that from which the backup was created. Then attempt to restore the NIM database backup.

Message 0042-235

Explanation An image source was not specified for creating the **SPOT**.

User Action Specify a device containing installation images or specify an **lpp_source** with the **simages** attribute for creating the **SPOT**.

Message 0042-236

Explanation A name for the **lpp_source** and/or a directory to contain the **lpp_source** was not specified for the **lpp_source** that will be created.

User Action Specify a name and a directory for the **lpp_source** and try the operation again.

Message 0042-237

Explanation A name for the **SPOT** and/or a directory to contain the **SPOT** was not specified for the **SPOT** that will be created.

User Action Specify a name and a directory for the **SPOT** and try the operation again.

Message 0042-238

Explanation A parent directory was not specified for the diskless and dataless machine resources that will be created.

User Action Specify a directory for the diskless/dataless machine resources and try the operation again.

Message 0042-239

Explanation A name for the resource and/or directory to contain the resource was not specified for the resource that will be created.

User Action Specify a name and a directory for the resource and try the operation again.

Message 0042-240

Explanation A parent directory was not specified for the diskless and dataless machine resources that will be created.

User Action Specify a directory for the diskless/dataless machine resources and try the operation again.

Message 0042-241

Explanation The size and/or volume group was not specified for the creation of a new file system to contain a NIM resource.

User Action Specify both the size and volume group for the file system and try the operation again.

Message 0042-242

Explanation The size and/or volume group was not specified for the creation of a new file system to contain diskless and dataless machine resources.

User Action Specify both the size and volume group for the file system and try the operation again.

Message 0042-243

Explanation An attempt was made to create the same file system twice: once for an **lpp_source** and once for a **SPOT**.

User Action Specify a different directory for either the **lpp_source** or the **SPOT**. This will cause different file systems to be created for the resources. If a new file system really should be created to contain both resources, then only specify that the file system should be created for one of the resources, but specify the same directory for both resources.

Message 0042-244

Explanation An attempt was made to create the same file system twice: once for an **lpp_source** and once for diskless/dataless machine resources.

User Action Specify a different directory for either the **lpp_source** or the diskless/dataless resources. This will cause different file systems to be created for the resources. If a new file system really should be created to contain both sets of resources, then only specify that the file system should be created for one of the resources, but specify the same directory for both resources.

Message 0042-245

Explanation An attempt was made to create the same file system twice: once for a **SPOT** and once for diskless/dataless machine resources.

User Action Specify a different directory for either the **SPOT** or the diskless/dataless resources. This will cause different file systems to be created for the resources. If a new file system really should be created to contain both sets of resources, then only specify that the file system should be created for one of the resources, but specify the same directory for both resources.

Message 0042-246

Explanation Not enough space on the volume group to create the specified file system.

User Action Specify a different volume group for the file system to be created and try the operation again.

Message 0042-247

Explanation Creation of the file system failed.

User Action Check the previous output for error messages to understand what caused the file system creation to fail. Correct the error and try the operation again.

Message 0042-248

Explanation An error occurred during file system creation.

User Action Check the previous output for error messages to understand what caused the file system creation to fail. Correct the error and try the operation again.

Message 0042-249

Explanation NIM master initialization failed.

User Action Check the previous output for error messages to understand what caused the configuration of the NIM master to fail. Correct the error and attempt to reinitialize the master. The most frequent cause of this failure is that the master is already initialized. The master can be unconfigured with the **nim -o unconfig master** command and reinitialized. However, this should be done with extreme caution, since unconfiguring the master will remove all definitions from the NIM database.

Message 0042-250

Explanation Unable to continue with configuration.

User Action Check the previous output for error messages to understand what caused the configuration to fail. Correct the error and attempt to configure the system again from the point of failure.

Message 0042-251

Explanation A route cannot be added to the network, because a required default route is missing. Add a default route to the network, and try this operation again.

User Action Add a default route to the network specified in the error message, and retry the operation.

Message 0042-252

Explanation Unable to locate a matching network.

User Action The **find_net** keyword was used in the **if** attribute of the machine. However, no matching network was found. Either define the network prior to defining the machine interface, or use the **net_definition** attribute in conjunction with the **find_net** keyword to define the network while the interface is being defined.

Message 0042-253

Explanation You cannot use the **net_definition** attribute when the **find_net** keyword is not specified as the first field of the **if** attribute.

User Action The **net_definition** attribute is invalid when using a known network in the **if** attribute. Specify the **find_net** keyword in the **if** attribute, or omit the **net_definition** attribute, and retry the operation.

Message 0042-254

Explanation Invalid format for the specified value of **net_definition**. The value of the attribute should be as follows:

NetType
: Network type (for example, `tok`, `ent`, `fddi`, etc.).

snmName
: Dotted decimal subnet mask for the network.

Client_gwName
: Optional default gateway IP address or host name used by the machine being defined to communicate with the master.

Master_gwName
: Optional default gateway IP address or host name used by the master to communicate with clients on other subnets.

NetName
: Optional name given to the NIM definition created for the network. (Otherwise, a unique default name is used.)

If you want to specify *NetName* and if *Client_gwName* or *Master_gwName* are not applicable, specify **0** in their place. If *Client_gwName* is **0**, *Master_gwName* cannot be nonzero.

User Action Correct the syntax error, and retry the operation.

Message 0042-255

Explanation The master already has a default route, and the gateway you specified as being the default for the master is different from that which is already defined. Use the **change** operation if you want to modify the master's default gateway.

User Action To change the default gateway for a network, use the following command:

```
nim -o change -a routingX="default GtName" NetName
```

where *X* is the sequence number for the **routing** attribute; *GtName* is the default gateway to use; and *NetName* is the name of the master's network.

Message 0042-256

Explanation A default route already exists for the network. You can modify the default gateway, but you cannot define more than one default route.

User Action To change the default gateway for a network, use the following command:

```
nim -o change -a routingX="default GtName" NetName
```

where *X* is the sequence number for the **routing** attribute; *GtName* is the default gateway to use; and *NetName* is the name of the network to modify.

Message 0042-257

Explanation You cannot specify the **net_definition** attribute without specifying the **if** attribute when changing a machine definition.

User Action The **net_definition** must reference a machine interface, so specify an **if** attribute when using the **net_definition** attribute.

Message 0042-258

Explanation You cannot specify the **net_definition** attribute when creating or modifying more than one if attribute in the same **change** operation. Use two separate operations.

User Action To avoid ambiguity, manipulate only one machine interface (**if** attribute) at a time when using the **net_definition** attribute.

Message 0042-259

Explanation The value of **default_res** specified on the master's database definition is not a valid NIM resource group.

User Action Specify a valid NIM resource group as the default resource. Obtain a list of resource groups by running the **lsnim -t res_group** command.

Message 0042-260

Explanation The **default** attribute is only applicable when manipulating a resource group.

User Action Setting the **default=yes/no** attribute on a resource group makes it the default set of resources to use in NIM operations. The **default** attribute is invalid when used as an attribute in other NIM operations.

Message 0042-261

Explanation Illegal use of the **async** attribute. This attribute can only be specified for the **lppchk** operation when the target is a standalone machine or a group of standalone machines.

User Action Omit the **async** attribute when performing the **lppchk** operation, unless the target is a standalone machine or a group of standalone machines.

Message 0042-262

Explanation The file name of the client definition file is missing for this operation.

User Action Specify the client definition file that should be used to add machines to the NIM environment. For more information, see"Chapter 9. Network Installation Management Commands" on page 123.

Message 0042-263

Explanation The **netboot_kernel** attribute can only be assigned a value of **up** or **mp**.

User Action Correct the value specified for the **netboot_kernel** attribute.

Message 0042-264

Explanation The image source that was used to define the **lpp_source** is missing one or more requested packages.

User Action Installation images were not copied into the **lpp_source** directory. The source for installation images may not contain all of the filesets specified to populate the **lpp_source**. Copy the missing installation images to the **lpp_source** directory, and then perform the NIM **check** operation on the **lpp_source**.

Message 0042-265

Explanation The image source that was used to define the **lpp_source** is missing one or more items from the list of default packages.

User Action Installation images were not copied into the **lpp_source** directory. The source for installation images may not contain all of the default filesets used to populate the **lpp_source**. Copy the missing installation images to the **lpp_source** directory, and then perform the NIM **check** operation on the **lpp_source**.

Message 0042-266

Explanation Requested packages are missing from the defined **lpp_source**.

User Action Installation images were not copied into the **lpp_source** directory. The fileset names may have been specified incorrectly, or the source for installation images may not contain all of the specified filesets. Copy the missing installation images to the **lpp_source** directory, and then perform the NIM **check** operation on the **lpp_source**.

Message 0042-267
Explanation The defined **lpp_source** does not have the **simages** attribute, because one or more packages are missing.
User Action Copy the missing installation images to the **lpp_source** directory, and perform the NIM **check** operation on the **lpp_source** to add the **simages** attribute.

Message 0042-268
Explanation The operation cannot be performed, because all members of the target group specified are currently excluded from operations on the group. You must unmark (or include) excluded group members before proceeding.
User Action Perform the NIM **select** operation on the group to include members in further operations.

Message 0042-269
Explanation Only one type of verification can be performed at a time when verifying installed filesets on a NIM client.
User Action Disable or deselect all but one verification option and try the operation again.

Message 0042-270
Explanation The operation is only supported on **SPOTs** and NIM clients installed with a version and release level of AIX 4.2 or greater.
User Action The NIM client fileset on the target is at an earlier level and does not support the attempted operation. The client software on the target must be upgraded before the operation can be performed.

Message 0042-271
Explanation A resource matching the type is already allocated. You cannot allocate more than one resource of this type to a machine.
User Action Deallocate the first resource before attempting to allocate the second. It may be necessary to reset the machine before the resource can be deallocated.

Message 0042-272
Explanation A value specified is not a valid value for **default_re** because it is not a valid NIM resource group.
User Action Specify a different resource group for the **default_res** attribute, or correct the resource group in question.

Message 0042-273
Explanation A value specified cannot be used as the location for the **mksysb** image because it is a directory. You must specify the filename where the **mksysb** image currently resides or will reside after creation.
User Action Specify a file name instead of a directory for the location of the **mksysb** resource.

Message 0042-274
Explanation The **-e** flag in the **mksysb_flags** attribute and the **exclude_files** attribute cannot be specified together. Specify the **-e** flag with the **mksysb_flags** attribute to exclude the files in **/etc/exclude.rootvg** from the backup, or specify an **exclude_files** attribute.
User Action Do not specify both the **-e mksysb** flag and an **exclude_files** resource when performing this operation.

Message 0042-275
Explanation Unable to obtain possession of a lock file. If no NIM operations are currently in progress, remove the file and repeat the operation.

User Action Use the **ps -ef | grep nim** command to list the running NIM processes on the system. If any NIM processes other than the **nimesis** daemon are running, wait for them to finish and then remove the file specified by the error message.

Message 0042-276

Explanation A fileset must be installed before this operation can be performed.

User Action Install the fileset listed in the error message before retrying the operation. Generally, the fileset needs to be installed on the client system. However, depending on the operation being performed, the NIM master may also need to have the fileset installed before the operation will succeed.

Message 0042-277

Explanation Diskless and dataless machines cannot be defined with a primary network install interface residing on a generic NIM network. It is presumed that a network adapter defined on a generic NIM network does not support network boot.

User Action To define the systems as diskless or dataless clients, they must first be connected to a NIM network that is known to support network boot, such as ethernet, token-ring, or FDDI.

Message 0042-278

Explanation The interface specified does not correspond to a network adapter that is known to support network boot. As a result, the NIM master has been defined on a generic NIM network. Network boot-dependent operations, such as base operating system installation, will not be possible on any NIM client whose primary network install interface is defined on the same network as the NIM master.

User Action Operations that rely on network boot capability cannot be performed on clients on generic NIM networks. Such operations must be performed using local media on the system.

Message 0042-279

Explanation The interface specified maps to a subnet which has been defined as a generic NIM network. It will not be possible to perform network boot-dependent operations, such as base operating system installation, on the machine definition created by this operation.

User Action Operations that rely on network boot capability cannot be performed on clients on generic NIM networks. Such operations must be performed using local media on the system.

Message 0042-280

Explanation Specify a complete date and time for the scheduled operation in the form: `YYMMDDhhmm`.

User Action Use the format described in the error message to correctly schedule a date and time for the operation.

Message 0042-281

Explanation The **/usr** file system on the specified server cannot be converted to a NIM **SPOT**. Either the **RM_INST_ROOTS** variable was set to **yes** in a **bosinst.data** file during initial installation of the machine or **inurid -r** was subsequently invoked. The only way to create a **SPOT** on this machine is to specify the location to be something other than **/usr** or reinstall the machine and then create a **SPOT** in **/usr**.

User Action The system is unable to support the creation of a **/usr SPOT**. A **non-/usr SPOT** may be created on the system by specifying a different value for the **location** attribute.

Message 0042-282

Explanation The BOS installation has been enabled but could not be initiated, because the following file was not found on the target. To start the installation, do one of the following:

1. Initiate a network boot operation from the target.
2. Correct the state of the target with NIM's **reset** operation and invoke the **bos_inst** operation again using one of the following:
   a. The Force Push option (**-a force_push=yes**)
   b. After installing and configuring the **bos.sysmgt.nim.client** fileset on the target.

User Action The NIM client fileset is not properly installed and configured on the target system. Follow the directions specified in the error message to correct the problem.

Message 0042-283

Explanation The existence of a file on the server indicates that a NIM **SPOT** may still be mounted in a subdirectory which will be removed by this operation. Before attempting the operation again, unmount the **SPOT**'s directory along with any other directories that may be mounted beneath the directory being removed. FAILURE TO DO SO WILL RESULT IN LOSS OF DATA ON THE **SPOT** SERVER!

User Action A **SPOT** operation failed, and NIM was unable to unmount all the directories mounted into the **SPOT**. Manually unmount the directories specified in the error message before retrying the operation. The **mount** command can be used to list the directories mounted on the system, and the **unmount** command can be used to unmount directories. Use the **-f** option with the **unmount** command if necessary to force the unmount.

Chapter 11. Network Installation Management Troubleshooting

This chapter suggests solutions for network boot problems and describes procedures for producing debug output for NIM BOS installations. Refer to “Chapter 10. Error and Warning Messages” on page 125 for information about error messages.

Debugging a Network Boot Problem

If a client machine is unable to network boot from its boot server, there may be a problem in one or more of the following stages of the network boot:

- “Establishing Network Communication Between the Client and Server”
- “Obtaining the Boot Image from the Server”
- “Running the Boot Image on the Client” on page 154.

The following sections describe steps that can be followed to determine the source of failures during each stage.

Warning: Temporary Level 3 Header

Establishing Network Communication Between the Client and Server

1. Before initiating the network boot on the client, perform a ping test from the client **bootp** menus.
2. If the ping test fails, verify that the client, server, and gateway addresses are specified correctly.
3. If the addresses are correct, try to ping the server from a different machine in the client’s subnet.
4. If the server can be pinged from another machine, the network adapter on the boot client may be faulty.
5. If the server cannot be pinged from another machine in the client’s subnet, there may be routing problems between the client and the server, or network communications on the server may be faulty. Perform network debugging procedures to determine the source of the problem.

Obtaining the Boot Image from the Server

1. If the ping test is successful, perform a network boot of the client. When a network boot is initiated on a client, a **bootp** request packet is sent from the client to the server. The server then replies with a packet to the client. The client machine displays the number of packets sent and received for the **bootp** request. If a packet is sent from the client, but none is received, another packet will be sent.
2. If **bootp** packets continue to be sent but not received, the boot server may not be responding to the request.
3. From the **bootp** server, view the **/etc/bootptab** file on the server. It should contain an entry for the client machine with the following information:

   ```
   <hostname of client>
   bf=<boot file>
   ip=<client ip address>
   ht=<network type>
   sa=<boot server address>
   sm=<client subnet mask>
   ha=<network adapter hardware address> (required only if bootp
   requests are sent by broadcasting)
   ```

   If an entry does not exist, either the NIM command used to set up the current operation failed, or the machine was reset before the boot operation could occur. Rerun the NIM **bos_inst**, **diag**, **or maint_boot** operation to prepare the server for the client boot request.

   If the entry exists in **/etc/bootptab**, verify that the specified data is correct. If a field contains incorrect data, the information that was used to define the machine or network in the NIM database was

probably wrong. Correct this problem by resetting the client machine, correcting the invalid data in the client or network definition, retrying the NIM operation, and rebooting the client.

4. If the **/etc/bootptab** file is correct, verify that the **inetd** daemon is running. If it is not running, start it and retry the network boot from the client. If the **inetd** daemon is running, it should automatically start the **bootpd** daemon when the **bootp** request is received at the server.
5. If the **bootpd** daemon is not started, verify that the **bootps** entry in the **/etc/inetd.conf** file is not commented out. If it is commented out, uncomment it and restart **inetd** with the **refresh -s inetd** command. Retry the network boot from the client.
6. If a **bootp** reply is still not received at the client, manually start the **bootpd** daemon in debug mode:
   a. Comment out the **bootps** entry from the **/etc/bootptab** file on the server.
   b. Stop all running **bootpd** processes.
   c. Restart **inetd** using the **refresh -s inetd** command.
   d. Start **bootpd** from the command line using the **/usr/sbin/bootpd -s -d -d -d** command.
7. Retry to network boot from the client. If no output is displayed from the running **bootpd** command, the client **bootp** request is not reaching the server. Verify that the addresses specified in the **bootp** menus are correct. If they are correct, perform network debugging procedures to determine why the packet is not reaching the server.
8. If the server receives the client **bootp** request, the running **bootpd** command displays output matching the client data in the **/etc/bootptab** file. Verify that the specified addresses are correct. This information is sent back to the client in the **bootp** reply.
9. If the client is still not receiving the **bootp** reply, perform network debugging procedures to determine why the reply packet is not reaching the client.
10. After the client receives the **bootp** reply, it will **tftp** the boot image from the server.
11. The number of **tftp** packets transferred to the client will be displayed at the client machine.
12. The boot image has been successfully retrieved at the client machine when the LED shows 299 on **rs6k**-platform machines or when the bottom third of the screen turns gray on other platform machines.
13. If the **tftp** of the boot image does not complete successfully, the client may be trying to get the wrong boot image. Verify that the client definition in the NIM database shows the correct platform and kernel type. If the data is incorrect, correct it, reset the client machine, rerun the NIM operation, and reboot the client over the network.
14. Verify that the **/tftpboot** directory on the boot server contains a link with the client name to the correct boot image. If the link does not exist, reset the client machine, rerun the NIM operation, and reboot the client over the network.
15. If the link with the client name is pointing to the correct boot image and the **tftp** of the boot image does not complete successfully, the boot image may be corrupted. Recreate the boot image by performing a NIM **check** operation with the **force** flag on the **SPOT**. If the client is not an **rs6k**-platform machine, also make sure the client has the latest version of the firmware installed.

Running the Boot Image on the Client

After the client machine has successfully received the boot image from the server, the most common errors encountered are hangs with the LED showing 608, 611, or 613. Some machines may not have LED displays. Debugging such problems on these machines will require using debug-enabled boot images. For information on building debug boot images, see “Producing Debug Output from the BOS Install Program” on page 157.

608
Explanation **tftp** retrieve of client info file failure.

Action If a 608 hang is encountered, verify that the *ClientName*.info file exists in the **/tftpboot** directory. If it does not exist, retry the NIM operation to create it. If it does exist, verify that **tftp** access to the **/tftpboot** directory is not restricted in the **/etc/tftpaccess.ctl** file. It is also possible that the network adapter was not configured properly in the boot environment. Use debug-enabled network boot images to look for errors in the boot environment. If the client is not an **rs6k**-platform machine, make sure that it has the latest version of firmware installed.

611

Explanation Remote mount of NFS file system failure.

Action 611 hangs occur when the client machine is unable to mount a resource from a server. Ensure that NFS is running on the resource server. Verify that the resources specified for the operation are exported properly by checking the **/etc/exports** and **/etc/xtab** files on the server. Also, confirm that the resources have permissions set correctly for reading. Debug-enabled network boot images can also be used to determine exactly which **mount** command is failing on the client.

613

Explanation Failure setting up route tables.

Action 613 hangs usually occur because a route is incorrectly defined for a network in the NIM database. Verify that the correct gateways are specified between networks, and all gateways are functional. Use debug-enabled network boot images to determine which routes could not be defined.

Producing Debug Output for NIM BOS Installations

Due to problems in the network or in the NIM configuration, clients may fail to boot properly and/or install. When this happens, it may be necessary to produce debug information in order to determine the cause of the problem. If a client machine fails to configure properly from the network boot image, debug output from the boot image can be obtained by building the debug-enabled image and attaching a tty to the client system. This will display the commands and output that are run as the client is configured before further processing is done by AIX.

If the system has been 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. The commands and output from the BOS install program will automatically be displayed on the tty if the boot image was built debug-enabled. If the boot image was not built for debugging, output can be obtained by either setting a value in a **bosinst.data** file or by entering special codes at the installation menus.

When problems arise during a NIM BOS installation, you will most likely get 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 network adapter or an inability to perform an operation from the client to the server. By examining the debug output, you can see what failed and make corrections to avoid the error in the future.

You will see the **showled** command running in the debug output. This command displays status values on the LEDs on the front of the machine. Frequently, known problems and solutions are referenced by the LED value that is displayed when a problem occurs. Some machines do not have LEDs for displaying such information. Therefore, when debugging problems on such machines, special attention should be given to observing the values that the **showled** commands are displaying.

Obtaining debug information from a network installation can save you time in determining the root cause of a problem. Usually, the problem will be an incorrect definition in the NIM environment that can be found without the debug information. However, with the debug information, you can significantly reduce the scope of the investigation.

Producing Debug Output from a Network Boot Image

1. Create debug versions of the network boot images using the **check** operation from the Web-based System Manager or SMIT interfaces or by using the following command:

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

   where `SPOTName` is the name of your **SPOT**.
2. Get the address for entering the debugger:

   From Web-based System Manager

   a. From the NIM Resources container, double-click the **SPOT** resource to open the properties notebook.

   b. Click the Boot Image Information tab in the properties notebook to obtain the address.

   Alternatively, you can use the following command to get the address:

   ```
   lsnim -a enter_dbg SPOTName
   ```

   where `SPOTName` is the name of your **SPOT**. The displayed output will be similar to the following:

   ```
   spot1:
      enter_dbg = "chrp.mp 0x001840d4"
      enter_dbg = "chrp.up 0x00160b7c"
      enter_dbg = "rs6k.mp 0x001840d4"
      enter_dbg = "rs6k.up 0x00160b7c"
      enter_dbg = "rspc.mp 0x001840d4"
      enter_dbg = "rspc.up 0x00160b7c"
   ```

   Write down the **enter_dbg** address for the client you are going to boot. For example, if your client is an **rs6k**-uniprocessor machine, you would write down the address `160b7c`.
3. Attach a tty device to your client system (port 1).
4. Set up and perform the NIM operation that will require the client to boot over the network. Boot the client over the network.
5. After the client gets the boot image from the **SPOT** server, the debug screen will appear on the tty. At the > prompt, enter:

   ```
   st Enter_dbg_Value 2
   ```

   where `Enter_dbg_Value` is the number you wrote down in step 2 as your machine type's **enter_dbg** value. Specifying a 2 at the address of the **enter_dbg** value prints the output to your tty.
6. Type `g` for `go` and press Enter to start the boot process.
7. Use Ctrl-s to temporarily stop the process as you watch the output on the tty. Use Ctrl-q to resume the process.
8. To rebuild your boot images in non-debug mode, use the following command:

   ```
   nim - Fo check SPOTName
   ```

   where `SPOTName` is the name of your **SPOT**.

   If the boot image is left in debug mode, 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.

Producing Debug Output from the BOS Install Program

There are two ways to obtain debug output from the BOS install program. Method A involves entering a special value at one of the installation menus. “Method B: To Produce Debug Output When Using a bosinst_data Resource” uses a **bosinst_data** resource to tell the installation program to display debug output.

Method A: To Produce Debug Output When Not Using a bosinst_data Resource

1. To enable debugging for the BOS install program, start by performing all the processing you would normally do to install a client.
2. Since you are not using a **bosinst_data** resource, you will be prompted to supply information about the installation to the BOS install program.
3. Select your console.
4. Select your language.
5. The **Welcome to Base Operating System Installation and Maintenance** menu is displayed. Instead of selecting one of the options, enter **911** at the prompt and press Enter.
6. Continue the normal procedure for selecting options and specifying data until the installation begins. Debug output will be sent to the client’s display as the installation proceeds.

Method B: To Produce Debug Output When Using a bosinst_data Resource

1. To enable debugging for the BOS install program, set the value `BOSINST_DEBUG = yes` in the `control_flow` stanza of the **bosinst.data** file that you are using for your **bosinst_data** resource.

   A minimum **bosinst.data** file for debugging purposes would contain the following lines:

   ```
   control_flow:
        BOSINST_DEBUG = yes
   ```
2. In addition to the processing you would normally do to install a client, include the modified **bosinst_data** resource as a resource for the operation. After the client boots over the network, it will use the **bosinst_data** resource to obtain settings for the installation. If the only data specified in your **bosinst.data** file is `BOSINST_DEBUG = yes`, you will be prompted for the remaining required information before the installation will continue. Debug output will be sent to the client’s display as the installation continues.

Port Conflicts with NIM and other Applications

When the NIM Master is configured, two port numbers are selected to be used by the nimesis daemon for client communications. The default port numbers are 1058 and 1059. If either port is taken by another application, the nimesis daemon will not run and nimclient commands will fail with an error similar to the following:

0042-006 nimclient: (To master) rcmd connection refused

If the nimesis daemon cannot be started, it may be necessary to stop the other applicants on the system to free the port.

Rebooting the system will usually eliminate the problem, because when a machine is booted, the nimesis daemon is started very early by **init** and the likelihood that the ports are taken will be very small.

Appendix A. Network Boot

Older model **rs6k**-platform machines may require IPL ROM emulation to boot over the network. To determine whether or not a running **rs6k** machine requires emulation, enter the command `bootinfo -q AdapterName` where `AdapterName` is the network adapter over which the client will be installed. If the adapter is network-boot enabled, the **bootinfo** command will return 1, and no emulation is required. For example, enter:

```
bootinfo -q tok0
```

or

```
bootinfo -q ent0
```

Note that in the examples, `tok0` and `ent0` are adapter names. Do not use network names, such as `tr0`, `en0`, or `et0`.

If an **rs6k** machine is not running, it is possible to determine whether IPL ROM emulation is required by booting the machine with the key turned to Secure. If the LEDs on the front of the machine eventually stop at 200, no emulation is needed.

> **Note:** On model numbers 570, 580, 58H, 59H, 591, and 595, the system powers on in the Secure mode, the power-on light does not come on, and the LED remains blank. Refer to the documentation that came with your particular hardware model for more information.

To create IPL ROM emulation, see “Creating IPL ROM Emulation Media” on page 162 . If you are using NIM on machines that must use IPL ROM emulation to boot from a network adapter, you should always initiate the boot once from the IPL menus from a system console on the client. This writes the necessary addresses to NVRAM, avoiding a problem with seeing alternating LEDs 227 and 229 during the network boot on older hardware. After initiating the network boot/install from the client via this method once, the problem is automatically corrected and subsequent network boot/installation may be intiatied from the NIM master.

The platform and kernel type of a client determines the procedure required to boot the machine over the network.

To determine the platform of a running machine, use the **bootinfo -p** command if the machine is running AIX Version 4.2 or later. If the machine is running AIX 4.1, use the **bootinfo -T** command.

To determine the kernel type of a running machine, use the **bootinfo -z** command.

Booting a Machine Over the Network

If you are booting an **rs6k** machine with an **up** kernel, use Method A. If you are booting an **rs6k** machine with an **mp** kernel, use Method B. For some models of **rspc** machines, you may use Method C. For all other platforms and kernel types, follow the procedures in your hardware documentation to perform the network boot.

Method A

1. Begin with your machine turned off.
2. If your client requires IPL ROM emulation, insert the emulation disk into the diskette drive of the client, and turn on the machine with the hardware key in the Service position. When the **bootp** menus display, continue with step 3.

If your client does not require emulation, turn the key to the Secure position and turn on the machine. Watch the LEDs on the front of the machine. They will eventually stop changing and display **200**. Then, change the key position to Service and quickly press the reset (yellow) button. When the **bootp** menus display, continue with step 3.

> **Note:** On model numbers 570, 580, 58H, 59H, 591, and 595, the system powers on in the Secure mode, the power-on light does not come on, and the LED remains blank. The system performs no further operations until the key is set to the Normal or Service position. Refer to the documentation that came with your particular hardware model for more information about performing a network boot of the machine.

3. From the **bootp** main menu, choose the Select BOOT (Startup) Device option.
4. In the next menu, select the boot device.

   Select the network adapter to be used. Choose the adapter with the correct network type (Ethernet, Token-Ring, etc.) and adapter characteristics (thick cable, twisted pair, 16 Mb data rate, 4 Mb data rate, etc.).
5. Set or change the network addresses.

   > **Note:** You do not need to type the ′.’ characters in the IP addresses, but you must specify any leading ′0’ characters that make up parts of the addresses.

   Specify the IP address of:
   - The client machine you are booting in the `client address` field.
   - Your **SPOT** server in the `bootp server address` field.
   - Your client’s gateway in the `gateway address` field.

   After you specify the addresses, enter **99** to save the addresses and return to the main menu. If no gateway is used by the client to communicate with the boot server, you can leave this field empty.

   If broadcasting will be used to obtain a boot image, leave these IP addresses empty.
6. From the main menu, select the Send Test Transmission (PING) option.
7. Verify that the displayed addresses are the same as the addresses you specified for your boot device.

   If the addresses are incorrect, enter **99** to return to the main menu. Then, go back to step 3.

   If the addresses are correct, select the START THE PING TEST option.

   If the ping test fails, verify that the addresses are correct, and perform network problem determination if necessary. If the ping test completes successfully, enter **99** to return to the main menu.
8. From the main menu, select the Exit Main Menu and Start System (BOOT) option.
9. Turn the hardware key to the Normal position, and press Enter to boot your client over the network.

Method B (Booting Micro Channel-Based, Symmetric Multiprocessor Systems)

1. Turn the key mode switch to the Secure position.
2. Turn the power switch on the system unit to the On position.
3. When the LED displays `200`, turn the key mode switch to the Service position.
4. Press the Reset button once.

   The LED persistently displays `260`, `261`, or `262`, and The Maintenance Menu screen appears.
5. Select the **System Boot** option on the Maintenance Menu screen.
6. Select the **Boot from Network** option from the sub-menu. The MAIN MENU is displayed.
7. Select the **Select BOOT (Startup) Device** option on the MAIN MENU screen.

8. Select the network adapter from which the machine will boot. If there are multiple network adapters installed, type 88 and press the Enter key to view the other entries. Type a number from the list and press the Enter key.

   Note: If you are using a Token-Ring network, select the number that corresponds to the correct ring speed for your environment.
9. If a network adapter is selected, the SET OR CHANGE NETWORK ADDRESSES screen is displayed next. The hardware address for the network adapter is displayed in the hardware address field. Record the hardware address for defining the NIM machine object.

   If this client and the BOOTP server are on the same LAN, leave the IP address fields as zeros for the BOOTP request to be broadcasted over the LAN. If there are multiple BOOTP servers on the LAN or the client is on a different network to the server, enter the client and server IP addresses. Type in the IP addresses using leading zeros to pad the network address fields, for example, 009.101.002.050. If this machine must use a gateway to reach the server, enter the IP address for the gateway.

   Type 99 and press the Enter key to save the address information and return to the MAIN MENU.
10. (This step is optional.) Select the **Send Test Transmission (PING)** option on the MAIN MENU to test the network connection between the client and the server systems. Type 3 and press the Enter key to start the ping test. If the ping test was not successful, check that the IP addresses are correct and that the physical network connections are sound. If the ping test was successful, type 99 and press the Enter key to return to the MAIN MENU.
11. Select the **Exit Main Menu and Start System (BOOT)** option.
12. Follow the instructions on the screen to turn the key mode switch to the Normal position and press the Enter key.

The BOOTP request will be issued, followed by a TFTP transfer of the network boot image.

Method C (Booting an rspc Platform Machine)

1. Begin with your machine turned off.
2. If your system requires a System Management Services (SMS) diskette, insert it into the diskette drive of the client and turn on the machine. If you do not insert an SMS diskette at this time and one is required, you will be prompted to insert one later.
3. A graphics image is displayed on your screen. Press the **F4** key as icons begin to display from left to right on the bottom of your display.

   Note: If the last icon is displayed prior to pressing the **F4** key, the normal mode boot list is used instead of the System Management Services diskette.
4. The System Management Services menu displays on your screen. Select the **Utilities** option.
5. From the System Management Services Utilities menu, select the **Remote Initial Program Load Setup** option.
6. From the Network Parameters screen, select the **IP Parameters option**.
7. Set or change the values displayed so they are correct for your client system.

   Note: You do not need to specify any leading ′0’ characters, but you must specify the ′.’ characters in the IP addresses.
8. Specify the IP address of:
   - The client machine you are booting in the `client address` field.
   - Your **SPOT** server in the `bootp server address` field.
   - Your client’s gateway in the `gateway address` field.

   Note: If broadcasting will be used to obtain a boot image, leave these IP address fields empty.

9. Specify the subnet mask for your client machine if you are prompted for one in the `subnet mask` field. All machines in your subnet have the same subnet mask.
10. After you specify the addresses, press Enter to save the addresses and continue.
11. The Network Parameters screen is displayed. Select the **Ping** option.
12. Select the network adapter to be used as the client's boot device.
13. Verify that the displayed addresses are the same as the addresses you specified for your boot device.
14. If the addresses are incorrect, press Esc until you return to the main menu. Then, go back to step 5.
15. If the addresses are correct, press Enter to perform the ping test. The ping test may take several seconds to complete.
16. If the ping test fails, verify that the addresses are correct, and perform network problem determination if necessary. If the ping test completes successfully, press Enter to acknowledge the *success* message. Then, press Esc until you return to the System Management Services menu.
17. From the System Management Services menu, choose the **Select Boot Devices** option.
18. Select the network adapter to be used for the network boot from the list of displayed bootable devices. Be sure to select the correct network type (Ethernet, Token-Ring, etc.) and adapter characteristics (thick cable, twisted pair, 16 Mb data rate, 4 Mb data rate, etc.). After making your selection, the machine will boot over the network.

    Note: When performing a BOS installation on a NIM client with a ″rspc″ platform, the machine may fail to boot from the network adapter if network traffic is heavy.

    If the network boot was intiated from the NIM Master, the machine will eventually boot from the disk. If the network boot was intiated from the SMS (System Management Services) menus on the NIM client, the machine will return control to the SMS menus.

    Contact your service representative to receive a firmware update to correct this problem.

Creating IPL ROM Emulation Media

Use this procedure to create the IPL ROM emulation media on the NIM master for machines that do not have a BOOTP-enabled IPL ROM.

1. Insert a formatted diskette or a tape into the appropriate drive on the NIM master.
2. Enter:

```
bosboot -T rs6k \
-r /usr/lpp/bos.sysmgt/nim/methods/IPLROM.emulation \
-d DeviceName -M both
```

   where *DeviceName* can be `fd0`, `/dev/fd0`, `rmt0`, or `/dev/rmt0`. This operation requires that the **devices.base.rte** fileset be installed on the machine upon which the emulation media is being created.
3. Insert the IPL ROM emulation media in the appropriate drive on the target machine.

From Web-based System Manager

1. From the NIM container, from the NIM menu, select **Create IPL ROM Emulation Media**.
2. Use the dialog to complete the task; all fields are required.

From SMIT

IPL ROM emulation can also be created using the **smit iplrom** fast path.

Appendix B. Hardware-Related Tasks

This chapter describes the procedure for determining a machine's network hardware address if broadcasting will be used to network boot the client.

Determining a Machine's Network Hardware Address

A machine's unique network hardware address associated with the network adapter is an optional part of the machine definition stored in the Network Installation Management (NIM) database. This address is used to configure the boot information for a client.

A value of zero may be specified as a network adapter hardware address when defining NIM machine objects. This is useful when a machine is not active and the hardware address cannot be obtained or when defining a large number of machines. If zero is used to specify a machine's network adapter hardware address, BOOTP requests initiated from the IPL ROM menus from a system console on the machine must have the IP address specified. A BOOTP request that broadcasts packets without an IP address will not work for a NIM client whose machine object has been defined with a network adapter address of zero.

If AIX has already been installed on the system, use one of the following procedures to determine the machine's network hardware address. If the system has not yet been installed or has been turned off, follow the procedure described in "Booting a Machine Over the Network" on page 159 for your specific platform. The network hardware address will be displayed during the setup procedure.

For a Running Machine

If TCP/IP has been configured and started, complete the following steps:

1. Enter:
   ```
   netstat -in
   ```
2. A table of information is displayed.
3. Find the section that applies to the interface you want to use as the machine's primary interface; for example, `tr0`. The hardware address of your machine's interface is listed after the entry `<Link>`. For example:
   ```
   tr0   1492  <Link>10.0.5a.a8.8f.94
   ```
4. Record the hardware address.

   Note: When you supply the network hardware address to NIM, which is part of defining a machine, delete the periods between the numbers. Also, `.0.` becomes`00`. For example, if the machine's network hardware address is displayed as `10.0.5a.a8.8f.94`, enter `10005aa88f94` when you define the machine.

If TCP/IP has not been configured, complete the following steps:

1. Enter `lscfg -l NetworkAdapterName -v`

   where *NetworkAdapterName* is the logical device name of the network adapter the client will use to access NIM resources. For example, `tok0`, `ent0` or `fddi0`.
2. A list of hardware information for the network adapter is displayed.
3. Find the field that starts with Network Address. This is the hardware address of your machine's interface. For example,
   ```
   Network  Address.............10005AA88F94
   ```
4. Record the hardware address.

Appendix C. Sample Files

This section covers the following sample files:

- **bosinst.data** File
- “script Resource File” on page 166
- “Definition File for the nimdef Command” on page 166

bosinst.data File

> **Note:** To check the contents of your customized **bosinst.data** file, use the **bicheck** command:
> `/usr/lpp/bosinst/bicheck` *`filename`*

The following is an example of a modified **bosinst.data** file. This file might be used in a nonprompted network installation. The default **bosinst.data** file that is supplied with the system includes detailed comments on the content and format of this file.

```
control_flow:
   CONSOLE = ( Device which will be your console. Passed to chcons. )
   INSTALL_METHOD = overwrite, preserve, migrate
   PROMPT = yes,no
   EXISTING_SYSTEM_OVERWRITE = yes,no,any
   INSTALL_X_IF_ADAPTER = yes,no,all
   RUN_STARTUP = yes,no
   RM_INST_ROOTS = yes,no
   ERROR_EXIT = (Command to run when error occurs. Must be in boot image.)
   CUSTOMIZATION_FILE = (Command to run after BOS install finishes.)
   TCB = yes, no  (turns on/off trusted computing base)
   INSTALL_TYPE = full (same as if blank), client, personal, eserver
   BUNDLES = (Fully qualified Bundle file(s) to install after auto install.
          Maximum of 139 bytes total. Custom bundle files
          provided on a diskette with the bosinst.data file must
          prefix the bundle file pathname with /../.
          If the list of bundle file names exceeds 139 bytes,
          cat all the bundle files together into a single custom bundle
          file and specify the name of the single custom bundle file.)
   RECOVER_DEVICES = yes,no
   BOSINST_DEBUG = no,yes
   ACCEPT_LICENSES =
   INSTALL_64BIT_KERNEL =
   INSTALL_CONFIGURATION =
   DESKTOP = CDE

target_disk_data:
   PVID = (physical volume identifier)
   CONNECTION = (parent//connwhere)
   LOCATION =
   SIZE_MB =
   HDISKNAME =

locale:
   BOSINST_LANG = (language to use during installation)
   CULTURAL_CONVENTION = (primary cc to use after reboot)
   MESSAGES = (primary message catalogs to use after reboot)
   KEYBOARD = (keyboard map to use after reboot)
```

> **Note:** In the control_flow stanza, **TCB** cannot be set to **yes** for SPOT installations (that is, when the **source** attribute is set to `spot`) or for **mksysb** installations if **TCB** was set to **no** when the **mksysb** image was created. For more information about the **bosinst.data** file, see the *AIX Version 4.3 Installation Guide*.

script Resource File

The following is an example of a customizing script that configures the target's TCP/IP domain name resolution and routing.

Manual customization is still necessary when configuring Domain Name Services for AIX 4.1 NIM clients. The **resolv_conf** resource should be used when installing clients running the latest version of AIX.

```
#!/bin/ksh  CUSTOMIZING SCRIPT to set the hostname,
#             establish the nameserver and DNS domain name,
#             and configure the routing table for the
#             target standalone client

# Truncate the host name
# if the host name is set to the fully qualified host name
#
#NOTE: This procedure will NOT result in a truncated host name if
#the bos installation operation is installing a mksysb image
#(ie. -a source=mksysb) unless the bos_inst operation is
#instructed not to configure the target as a NIM client upon
#completion (ie. unless -a no_nim_client=yes is specified)
#
chdev -l inet0 -a hostname =$(/usr/bin/hostname | cut -d. -f1)
# Set Name server and Domain Name

if [[ -f /etc/resolv.conf ]]
then
   /usr/sbin/namerslv -E '/etc/resolv.conf.sv'
fi
/usr/sbin/namerslv -a -i '9.101.1.70'
/usr/sbin/namerslv -c 'enterprise.ca'

# Flush routing table and add default route

/etc/route -n -f
odmdelete -o CuAt -q "name=inet0 and attribute=route"
chdev -l inet0 -a route=net,,'0','9.101.1.70'
```

Definition File for the nimdef Command

The following is an example of a definition file for the **nimdef** command:

```
# Set default values.

default:
    machine_type  = standalone
    subnet_mask   = 255.255.240.0
    gateway       = gateway1
    network_type  = tok
    ring_speed    = 16
    platform      = rs6k
    machine_group = all_machines

# Define the machine "lab1"
# Take all defaults.

lab1:
# Define the machine "lab2"
# Take all defaults and specify 2 additional attributes.
# The machine "lab2" uses IPL ROM emulation, and will be added to
# the machine groups "all_machines" (by default) and "lab_machines".

lab2:
    ipl_rom_emulation  = /dev/fd0
    machine_group      = lab_machines
```

```
# Define the machine "lab3"
# Take all defaults, but do not add the machine to the
# default group.

lab3:
    machine_group=

# Define the machine "lab4"
# Take all defaults, but do not add "lab4" to the default group
# "all_machines".
# Instead add it to the groups "lab_machines" and "new_machines".

lab4:
    machine_group =
    machine_group = lab_machines
    machine_group = new_machines

# Change the default "platform" attribute.

default:
    platform = rspc

# define the machine "test1"
# Take all defaults and include a comment.

test1:
    comments = "This machine is a test machine."
```

Appendix D. NIM Compatibility and Limitations

This section contains information about compatibility and limitations between AIX 4.1 and AIX 4.2 (and later).

Operations

showlog	Not supported on AIX 4.1 **SPOT** or machine targets.
showres	Not supported on AIX 4.1 **SPOT** or machine targets.
lppchk	Not supported on AIX 4.1 **SPOT** or machine targets.
maint_boot	Cannot use an AIX 4.1 **SPOT** as the **SPOT** resource when booting a client machine.

Resources

The **resolv_conf** resource is not supported when allocated to an AIX 4.1 machine target.

Resources served from an AIX 4.1 client cannot be used when performing operations on machine groups.

Creating a system backup image while defining a **mksysb** resource is not supported when using an AIX 4.1 client as the source of the backup.

Default Routes

Default routes cannot be used to install an AIX 4.1 machine from an AIX 4.2 (and later) master, unless APAR IX59393 is installed in the AIX 4.1 **SPOT** used for the installation.

NIM Master

Due to changes in the list of filesets to copy for **lpp_sources**, a later version **lpp_source** cannot be served from a NIM master running an earlier version of the operating system. An Authorized Program Analysis Report (APAR) may be applied to the AIX 4.1 NIM master to allow this function. Apply APAR IX59032 to the AIX 4.1 NIM master.

Due to kernel changes in AIX, a **SPOT** cannot be created at a level of AIX that is higher than the level of AIX installed on the server of the **SPOT**. For example, an AIX 4.1 NIM master cannot serve an AIX 4.2 (and later) **SPOT**. In order to have an AIX 4.2 (and later) **SPOT** in an environment where the NIM master is running AIX 4.1, a client may be installed with AIX 4.2 (and later), and it can serve the AIX 4.2 (and later) **SPOT**. For backwards compatibility, an AIX 4.2 (and later) master may serve AIX 4.1 **lpp_source** and **spot** resources.

Note: An older release cannot serve the **lpp_source** of a later release.

Memory Requirements for AIX 4.3

Machines must have at least 32 MB of memory in order to run AIX 4.3.

Remote Dump over ISA Ethernet

Remote dump is not supported over ISA ethernet. Therefore, an error will occur under either of these conditions:

- booting a diskless NIM client over ISA ethernet with a dump resource allocated.
- issuing the sysdumpdev command with a remote dump location specified as the argument.

This is the error message you will receive:

Cannot set primary dump device host:/location
Operation not supported on socket where host:/location is the remote location at which you are attempting to set the dump device.

LAN SCSI Adapter

On clients using a LAN SCSI adapter to access a NIM server it is not possible to declare the client with the niminit command.

It is possible to work around this problem by creating the NIM object that describes the client machine on the NIM server. This is done using the following command:

nim -o define

Appendix E. Notices

This information was developed for products and services offered in the U.S.A.
IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
IBM Corporation
Dept. LRAS/Bldg. 003
11400 Burnet Road
Austin, TX 78758-3498
U.S.A.
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and

cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Glossary

allocate. To make resources available to the *target* (page 174) machine.

attribute. A characteristic or property of one or more *objects* (page 174).

Base Operating System (BOS) installation. The process of installing and configuring the minimum amount of software needed to bring a machine to the running state.

boot image. An image containing the kernel, file systems, libraries, and programs. The boot image is loaded after the machine is turned on or reset and brings it to a running state.

client. A machine that uses remote *resources* (page 174).

configuration. (1.) The group of machines, devices, and programs that make up a data processing system or network. (2.) The process of describing to a system the devices, optional features, and program products that have been installed so that these features can be used. Contrast with *customization* (page 173).

connectivity. An algorithm that determines if two machines on different networks can communicate. If the machines can communicate, connectivity also determines which host names should be used and which TCP/IP routing information must be added.

control state. A state that represents the current Network Installation Management (NIM) operation being performed on a machine. This state is one of two *machine states* (page 173).

cross-platform resource. A resource that has an architecture different from the architecture of its server. Valid architecture types are POWER-based and Itanium-based.

customization. (1.) In the *Network Installation Management* (page 173) (NIM) environment, this is optional software installation. (2.) The process of describing optional changes to defaults of a software program that is already installed on the system and configured so that it can be used. Contrast with *configuration* (page 173).

fileset. One or more separately installable, logically grouped units in an installation package.

generic network. A Network Installation Management (NIM) network type used to define networks which are not currently known by NIM to support network boot operations.

groups. Network Installation Management (NIM) permits machines to be grouped together so that NIM operations may be easily directed at multiple *targets* (page 174). Similarly, NIM permits *resources* (page 174) to be grouped together and allocated as a logical unit to NIM targets.

machine execution state. A state that indicates the machine is shut down, booting, or running. This state is one of two *machine states* (page 173).

machine object. An entry in the Network Installation Management (NIM) database that represents a machine configuration.

machine state. A state that identifies the *machine execution state* (page 173) and *control state* (page 173) for each machine.

master. The only machine in the *Network Installation Management* (page 173) (NIM) environment that has permission to remotely execute commands on other NIM *clients* (page 173).

multihomed machine. A machine that has more than one configured network adapter and more than one host name.

network boot image. A *boot image* (page 173) that supports standalone, diskless, and dataless machines.

Network Installation Management (NIM). An environment that provides installation and configuration of software within a network interface.

network object. An entry in the Network Installation Management (NIM) database that represents a local area network.

network state. A state that indicates either that the *network object* (page 173) can participate in

Network Installation Management (NIM) operations or an error in the definition of the network object.

NIM routing. The information that defines which networks in the Network Installation Management (NIM) environment can communicate with each other and which gateways they use to facilitate that communication. NIM routing is used to represent the TCP/IP routing that exists for the LANs in the overall network environment.

objects. In the *Network Installation Management* (page 173) (NIM) environment, an entry in the NIM database that represents a machine, network, or *resource* (page 174).

pull installation. In the *Network Installation Management* (page 173) (NIM) environment, an installation that is initiated from a *target* (page 174).

push installation. In the *Network Installation Management* (page 173) (NIM) environment, an installation that is initiated from the master (page 173).

push permissions. Permissions that enable remote execution of commands.

resource. Any file, directory, file system, or device that is required to perform a *Network Installation Management* (page 173) (NIM) operation.

resource state. A state that indicates that the resource is either available or unavailable for use.

server. A machine that makes resources (page 174) available to other machines over a network.

Shared Product Object Tree (SPOT). For Network Installation Management (NIM), a **/usr** file system or an equivalent file system that is exported by servers in the *Network Installation Management* (page 173) environment for remote *client* (page 173) use.

software configuration. The processing required to make installed software ready to use.

software installation. The process of restoring software from external media to a local file system. The software can require further processing, or *configuration* (page 173), before it is ready to use.

target. For Network Installation Management, the *client* (page 173) you are installing.

Web-based System Manager. A Web-based graphical interface that allows you to perform installation and system management tasks.

Index

Special Characters

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

W

Readers' Comments — We'd Like to Hear from You

AIX 5L Version 5.1
Network Installation Management Guide and Reference

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Corporation
Publications Department
Internal Zip 9561
11400 Burnet Road
Austin, TX
78758-3493

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A