AIX Version 7.2

Installation and migration

AIX Version 7.2

Installation and migration

Note
Before using this information and the product it supports, read the information in "Notices" on page 439.

This edition applies to AIX Version 7.2 and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2015, 2018.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document v
Highlighting v
Case-sensitivity in AIX v
ISO 9000. v

Installation and migration 1
What's new in Installation and migration 1
Scenarios: Installing AIX 2
Installing new and complete overwrite BOS from media 2
Migrating your system from media 6
Creating and installing a software bundle. . . . 9
Adding open source applications to your AIX system 11
Cloning a rootvg using alternate disk installation 13
Configuring NIM using EZNIM 15
Installing a client using NIM 16
Network installation of a JS20 blade 17
Creating a system backup to tape 21
Cloning a system using a system backup tape . . 22
Cleaning up a failed software installation . . . 23
Installing AIX using the media device to install a partition with an HMC 24
Installing AIX using the media device to install a partition without an HMC 28
Configuring the AIX system after a new installation 31
Activation Engine 33
Installing the Base Operating System 39
Using BOS menus 41
Electronic license agreements 43
BOS installation options 44
The bosinst.data file 46
Installing new and complete BOS overwrite or preservation 57
AIX relocatable installation 62
Installing BOS on an iSCSI disk. 65
Installing BOS to an alternate disk. 67
Using the multibos utility 76
Customizing your installation 81
Installing content for the man command. . . . 85
Configuring AIX. 85
Configuring AIX with the Configuration Assistant 85
Configuring AIX with the Installation Assistant 86
Related information 87
Troubleshooting your installation 87
Troubleshooting an installation from a system backup 87
Troubleshooting migration installation 89
Troubleshooting alternate disk installation errors 90
Troubleshooting after a BOS installation 91
Troubleshooting a system that does not boot from the hard disk 91
Troubleshooting a full `/usr` file system 94
Viewing BOS installation logs 94
Interpreting installation-related system and error messages 95
Network Installation Management 107
NIM concepts 107
Configuring NIM 121
Installing with NIM 166
Setting up NIM networks 185
Booting with NIM. 191
Administering NIM 195
Managing NIM. 214
Using NIM resources 224
Using NIM operations 257
Using EZNIM 282
Using network installation files 284
Troubleshooting NIM. 288
Creating and installing system backups. 317
Creating system backups 318
Installing system backups 329
Optional products and service updates 335
Optionally installed software 335
Identifying software products 336
Software licensing 337
Managing AIX editions 337
Preparing to install optional software products and service updates 338
Checking fileset build dates 338
Installing optional software products or service updates 339
Maintaining optional software products and service updates. 343
Cleaning up optional software products and service updates. 345
Using the Software Service Management menu (including SUMA). 346
Using InstallShield MultiPlatform 350
Interim fix management solution 354
Live Update. 376
Software product packaging 400
Installing variously formatted software packages 400
Fileset installation packages 401
Creating software packages. 402
Packaging software bundles 403
Migrating AIX 405
AIX binary compatibility 408
BOS pre_migration and post_migration checks 410
Migrating to AIX Version 7.2 411
Migrating a multibos instance of AIX 414
mksysb migration. 415
Partitioning 420
Partitioning concepts 420
Implementations of logical partitions 422
Network adapter communication between partitions and the HMC 423
Installing AIX in a partitioned environment . . 424
Creating and changing a dedicate dump device 435

© Copyright IBM Corp. 2015, 2018

Verifying your dump device 436
Shutting down a partition 437
Changing your operating system host name . . 437

Notices 439
Privacy policy considerations 441
Trademarks 441

Index 443

About this document

This document provides system administrators with complete information about how to perform such tasks as installing and maintaining the AIX® operating system and optional software on standalone systems and on client systems from a resource server using the Network Install Management (NIM) interface. It also includes information on how to migrate a system, manage system backups, install AIX updates, use alternate disk installation, and troubleshoot problems with installation. This document is available on the documentation CD or DVD that is shipped with the operating system.

Highlighting

The following highlighting conventions are used in this document:

Item	Description
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.

Case-sensitivity in AIX

Everything in the AIX operating system is case-sensitive, which means that it distinguishes between uppercase and lowercase letters. For example, you can use the **ls** command to list files. If you type `LS`, the system responds that the command `is not found`. Likewise, **FILEA**, **FiLea**, and **filea** are three distinct file names, even if they reside in the same directory. To avoid causing undesirable actions to be performed, always ensure that you use the correct case.

ISO 9000

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

© Copyright IBM Corp. 2015, 2018

Installation and migration

This topic provides system administrators with complete information about how to perform such tasks as installing and maintaining the AIX operating system and optional software on standalone systems, and on client systems from a resource server using the Network Install Management (NIM) interface. It also includes information about how to migrate a system, manage system backups, install AIX updates, use alternate disk installation, and troubleshoot problems with installation. This topic is available on the documentation media that is shipped with the operating system.

What's new in Installation and migration

Read about new or significantly changed information for the Installation and migration topic collection.

How to see what's new or changed

In this PDF file, you might see revision bars (|) in the left margin that identify new and changed information.

September 2018

The following information is a summary of the updates made to this topic collection:

- Added information about updating the `lvupdate.data` file to run Live Update operations on another host in the "Configuring resources for Live Update" on page 384 topic.
- Added information about partitions being temporarily moved when certain circumstances apply in the Live Update best practices topic.

August 2018

The following information is a summary of the updates made to this topic collection:

- Added information about the "Using NIM for installing AIX updates and new packages over the HTTP protocol" on page 124 topic.
- Added a note about using IBM Spectrum Protect Space Management (HSM) during a AIX® Live Update operation in the "Live Update restrictions" on page 379 topic.

February 2018

The following information is a summary of the updates made to this topic collection:

- Updated restriction information for the AIX® Live Update operation in the "Live Update restrictions" on page 379 topic.

January 2018

The following information is a summary of the updates made to this topic collection:

- Updated information about using the Live Update function in PowreVC management in the "Best practices for the Live Update function" on page 382 topic.

December 2017

The following information is a summary of the updates made to this topic collection:

- Updated information about the **Trusted Excecution** option in the "Live Update restrictions" on page 379 topic.

© Copyright IBM Corp. 2015, 2018

October 2017

The following information is a summary of the updates made to this topic collection:

- Updated information about the Enterprise Pool CoD resources in the "Configuring resources for Live Update" on page 384 topic.
- Updated information about the `LVUP_COMPLETE` phase in the "Timeline to run the DLPAR scripts" on page 393 topic.
- Updated information about the Live Update support if the logical partition is managed by the PowerVC in the following topics:
  - "Defining NIM clients" on page 108"Adding PowerVC management objects to the NIM environment" on page 134
  - "Live Update" on page 376
  - "Best practices for the Live Update function" on page 382
  - "LPAR requirements for Live Update" on page 378
  - "Prerequisites for Live Update" on page 388
  - "Configuring resources for Live Update" on page 384
  - "Performing the Live Update operation by using NIM" on page 389
  - "Performing the Live Update operation by using the geninstall command" on page 390

Scenarios: Installing AIX

Use how-to's to perform common installation tasks.

Installing new and complete overwrite BOS from media

Using this scenario, you can install the AIX operating system for the first time or overwrite an existing version of the operating system.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

In this scenario, you will do the following:

- Boot from the AIX product media
- Set BOS Installation Settings
  - Perform a new and complete overwrite installation of AIX onto `hdisk0`
  - Use English as the primary language
  - Use the default options in the More Options menu
- Start the BOS Installation and Configure the System

If you are overwriting an existing system, gather the TCP/IP information from the system before you begin this scenario.

Attention: This procedure requires shutting down and reinstalling the base operating system. Whenever you reinstall any operating system, schedule your downtime when it least impacts your workload to protect yourself from a possible loss of data or functionality. Before you perform a new and complete overwrite installation, ensure you have reliable backups of your data and any customized applications or volume groups. For instructions on how to create a system backup, refer to Creating system backups.

The following steps show you how to use the system's built-in media device to perform a new and complete overwrite base operating system installation.

At this point, the BOS Installation is complete, and the initial configuration of the system is complete.

Step 1. Prepare your system

- There must be adequate disk space and memory available. The AIX operating system requires minimum of 4 GB of memory and 20 GB of physical disk space. For additional release information, see the *AIX 7.2 Release Notes*.
- Make sure your hardware installation is complete, including all external devices. See the documentation provided with your system unit for installation instructions.
- If your system needs to communicate with other systems and access their resources, make sure you have the information in the following worksheet before proceeding with installation:

Table 1. Network Configuration Information Worksheet

Network Attribute	**Value**
Network Interface	(For example: en0, et0)
Host Name	
IP Address	_____._____._____._____
Network Mask	_____._____._____._____
Nameserver	_____._____._____._____
Domain Name	
Gateway	_____._____._____._____

Step 2. Boot from the AIX product media

Booting the system from the AIX Product media.

1. Insert the *AIX Volume 1* media into the media device.
2. Make sure all external devices attached to the system (such as DVD drives, and terminals) are turned on. Only the media drive from which you will install AIX should contain the installation media.
3. Power on the system.
4. When the system beeps twice, press F5 on the keyboard (or 5 on an ASCII terminal). If you have a graphics display, you will see the keyboard icon on the screen when the beeps occur. If you have an ASCII terminal (also called a tty terminal), you will see the word `keyboard` when the beeps occur.

   Note: If your system does not boot using the F5 key (or the 5 key on an ASCII terminal), refer to your hardware documentation for information about how to boot your system from an AIX product media.
5. Select the system console by pressing F1 (or 1 on an ASCII terminal) and press Enter.
6. Select the English language for the base operating system (BOS) Installation menus by typing a 1 in the **Choice** field. Press Enter to open the Welcome to Base Operating System Installation and Maintenance screen.
7. Type 2 to select **2 Change/Show Installation Settings and Install** in the **Choice** field and press Enter.

```
                    Welcome to Base Operating System
                    Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

>>> 1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    5 Select Storage Adapters

    88  Help ?
    99  Previous Menu
>>> Choice [1]: 2
```

Step 3. Set and verify BOS installation settings

1. In the Installation and Settings screen, verify the installation settings are correct by checking the method of installation (new and complete overwrite), the disk or disks you want to install, the primary language environment settings, and the **more options** menu.

   If the default choices are correct, type `0` and press Enter to begin the BOS installation. The system automatically reboots after installation is complete. Go to step 4.

   Otherwise, go to sub-step 2.

2. To change the System Settings, which includes the method of installation and disk where you want to install, type 1 in the **Choice** field and press Enter.

```
                        Installation and Settings

Either type 0 and press Enter to install with current settings, or type the
 number of the setting you want to change and press Enter.

    1  System Settings:
         Method of Installation.............New and Complete Overwrite
         Disk Where You Want to Install.....hdisk0

>>> Choice [0]: 1
```

3. Type 1 for New and Complete Overwrite in the **Choice** field and press Enter. The Change Disk(s) Where You Want to Install screen now displays.

```
                Change Disk(s) Where You Want to Install

Type one or more numbers for the disk(s) to be used for installation and press
Enter. To cancel a choice, type the corresponding number and Press Enter.
At least one bootable disk must be selected. The current choice is indicated
by >>>.

        Name      Location Code   Size(MB)  VG Status   Bootable

     1  hdisk0   04-B0-00-2,0      30720   none            Yes
     2  hdisk1   04-B0-00-5,0      30720   none            Yes
     3  hdisk2   04-B0-00-6,0      12288   none            Yes

>>>  0   Continue with choices indicated above

    66  Disks not known to Base Operating System Installation
    77  Display More Disk Information
    88  Help ?
    99  Previous Menu

>>> Choice [0]:
```

4. In the Change Disk(s) Where You Want to Install screen:
   a. Select **hdisk0** by typing a 1 in the **Choice** field and press Enter. The disk will now be selected as indicated by >>>. To unselect the destination disk, type the number again and press Enter.
   b. To finish selecting disks, type a 0 in the **Choice** field and press Enter. The Installation and Settings screen displays with the selected disks listed under System Settings.
5. Change the Primary Language Environment Settings to English (United States). Use the following steps to change the Cultural Convention, Language, and Keyboard to English.
   a. Type 2 in the **Choice** field on the Installation and Settings screen to select the **Primary Language Environment Settings** option.
   b. Type the number corresponding to English (United States) as the Cultural Convention in the **Choice** field and press Enter.
   c. Select the appropriate keyboard and language options.

   You do not need to select the **More Options** selection, because you are using the default options in this scenario. For more information about the installation options available in AIX, see BOS installation options.
6. Verify that the selections are correct in the Overwrite Installation Summary screen, as follows:

```
     Overwrite Installation Summary

Disks:  hdisk0
Cultural Convention:  en_US
Language:  en_US
Keyboard:  en_US
Graphics Software: Yes
Desktop:  CDE
System Management Client Software:  Yes
OpenSSH Client Software:  No
OpenSSH Server Software:  No
Enable System Backups to install any system:  Yes
Selected Edition:  express

Optional Software being installed:

>>> 1   Continue with Install
    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

   The default options change based on machine, security, and console type.
7. Press Enter to begin the BOS installation. The system automatically reboots after installation is complete.

Step 4. Configure the system after installation

1. On systems with a graphics display, after a new and complete overwrite installation, the Configuration Assistant opens. On systems with an ASCII display, after a new and complete overwrite installation, the Installation Assistant opens.
2. Select the **Accept Licenses** option to accept the electronic licenses for the operating system.
3. Set the date and time, set the password for the administrator (root user), and configure network communications (TCP/IP).

   Use any other options at this time. You can return to the Configuration Assistant or the Installation Assistant by typing configassist or *smitty assist* at the command line.
4. Select **Exit the Configuration Assistant** and select Next. Or, press F10 (or ESC+0) to exit the Installation Assistant.
5. If you are in the Configuration Assistant, select **Finish now, and do not start Configuration Assistant when restarting AIX** and select **Finish**.

Remove disk reservations

If the disk you select to install to, is reserved by another system, the reservation can be removed.

The check for reservations is run only on disks you select to install, and if the reservation exists, an informational message is provided. During a non-prompted install if a reservation is detected on a selected disk the installation is changed to prompted, with an informational message.

How to use the Remove disk reservations menu

The **Remove disk reservations** menus can be accessed from the **Make Additional Disks Available** choice on the main menu of the **Base Operating System** menus. These menus are available for network installations and DVD installations only. When you are starting the system from a system backup tape, and must remove a disk reservation, select option `3` **Start Maintenance Mode for System Recovery** and then option `3` **Access Advanced Maintenance Functions**. The **devrsrv** command can be run at the shell prompt to remove and query disk reservations.

To access the **Remove disk reservations** menus, complete the following steps:

1. From the base operating system (**BOS**) menus, select **Welcome to Base Operating System Installation and Maintenance**.
2. Choose the **Make Additional Disks Available** option
3. Choose the **Remove disk reservation** option to access the **SMIT** interface to query and remove a reservation on a disk.
4. When you select the disk for the action, be sure to compare the `world wide name` and `LUN ID`, during the restart from installation media, the disk numbering may change.
5. After the action is complete select `F10` to exit back to **BOS** Menus.
6. Select **Change/Show Installation Settings and Install** to confirm your disk selection, and continue the installation.

Migrating your system from media

Using this scenario, you can migrate a system from AIX 6.1 to AIX 7.2.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

In this scenario, you will do the following:

- Perform a migration installation of AIX 6.1 to AIX 7.2.
- Use English as the primary language.
- Use the default options in the **Advanced Options** menu.

Attention: This procedure requires shutting down and reinstalling the base operating system. Whenever you reinstall any operating system, schedule your downtime when it least impacts your workload to protect yourself from a possible loss of data or functionality. Before you perform a migration installation, ensure you have reliable backups of your data and any customized applications or volume groups. For instructions on how to create a system backup, refer to Creating system backups in *Installation and migration*.

Step 1. Prepare for the migration

Before starting the migration, complete the following prerequisites:

- Ensure that the root user has a primary authentication method of SYSTEM. You can check this condition by typing the following command:

```
# lsuser -a auth1 root
```

If needed, change the value by typing the following command:

```
# chuser auth1=SYSTEM root
```

- Before you begin the installation, other users who have access to your system must be logged off.
- Verify that your applications will run on AIX 7.2. Also, check if your applications are binary-compatible with AIX 7.2. If your system is an application server, verify that there are no licensing issues. Refer to your application documentation or provider to verify on which levels of AIX your applications are supported and licensed.
- Check that your hardware microcode is up-to-date.
- All requisite hardware, including any external devices (such as tape drives or CD/DVD-ROM drives), must be physically connected and powered on. If you need further information, refer to the hardware documentation that accompanied your system.
- Use the **errpt** command to generate an error report from entries in the system error log. To display a complete detailed report, type the following:

  ```
  # errpt -a
  ```

- Adequate disk space and memory must be available. You need at least 4 GB of memory a minimum of 20 GB of physical disk space.
- Run the **pre_migration** script located in the ***mount_point***`/usr/lpp/bos` directory on your media. To mount the media, enter the following command, where *N* is your media drive number:

  ```
  # mount -v cdrfs -o ro /dev/cdN  /mnt
  ```

  Note: Do not remove the data created by the **pre_migration** script, because it is used by the **post_migration** script.
- For the latest migration information see the latest release notes.

Step 2. Boot from the AIX product media

1. If they are not already on, turn on your attached devices.
2. Insert the *AIX Volume 1* media into the media device.
3. Reboot the system by typing the following command:

   ```
   # shutdown -r
   ```

4. When the system beeps twice, press F5 on the keyboard (or `5` on an ASCII terminal). If you have a graphics display, you will see the keyboard icon on the screen when the beeps occur. If you have an ASCII terminal (also called a tty terminal), you will see the word `keyboard` when the beeps occur.

   Note: If your system does not boot using the F5 key (or the `5` key on an ASCII terminal), refer to your hardware documentation for information about how to boot your system from an AIX product media.
5. Select the system console by pressing F1 (or `1` on an ASCII terminal) and press Enter.
6. Select the English language for the BOS Installation menus by typing a `5` at the **Choice** field and press Enter. The Welcome to Base Operating System Installation and Maintenance menu opens.
7. Type `2` to select **2 Change/Show Installation Settings and Install** in the **Choice** field and press Enter.

```
                    Welcome to Base Operating System
                    Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

    1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    5 Select Storage Adapters


    88  Help ?
    99  Previous Menu
>>> Choice [1]: 2
```

Step 3. Verify migration installation settings and begin installation

1. Verify that migration is the method of installation. If migration is not the method of installation, select it now. Select the disk or disks you want to install.

```
    1  System Settings:
         Method of Installation.............Migration
         Disk Where You Want to Install.....hdisk0
```

2. Select **Primary Language Environment Settings (AFTER Install)**.
3. Type 3 and press Enter to select **More Options**. To use the Help menu to learn more about the options available during a migration installation, type 88 and press Enter in the Installation Options menu. For more information about the installation options available in AIX 7.2, see BOS installation options.
4. Verify the selections in the Migration Installation Summary screen and press Enter.
5. When the **Migration Confirmation** menu opens, follow the menu instructions to list system information or continue with the migration by typing 0 and pressing Enter.

```
                                Migration Confirmation

  Either type 0 and press Enter to continue the installation, or type the
  number of your choice and press Enter.

    1  List the saved Base System configuration files which will not be
       merged into the system.  These files are saved in /tmp/bos.
    2  List the filesets which will be removed and not replaced.
    3  List directories which will have all current contents removed.
    4  Reboot without migrating.

    Acceptance of license agreements is required before using system.
    You will be prompted to accept after the system reboots.

>>> 0  Continue with the migration.
   88  Help ?

+-------------------------------------------------------------------------
  WARNING: Selected files, directories, and filesets (installable options)
    from the Base System will be removed.  Choose 2 or 3 for more information.


>>> Choice[0]:
```

Step 4. Verify system configuration after installation

After the migration is complete, the system will reboot. as follows:

1. On systems with a graphics display, after a migration installation, the Configuration Assistant opens. On systems with an ASCII display, after a migration installation, the Installation Assistant opens.

   For more information on the Configuration Assistant or the Installation Assistant, see Configuring AIX with the Configuration Assistant.
2. Select the **Accept Licenses** option to accept the electronic licenses for the operating system.
3. Verify the administrator (root user) password and network communications (TCP/IP) information.

   Use any other options at this time. You can return to the Configuration Assistant or the Installation Assistant by typing `configassist` or `smitty assist` at the command line.
4. Select **Exit the Configuration Assistant** and select **Next**. Or, press F10 (or `ESC+0`) to exit the Installation Assistant.
5. If you are in the Configuration Assistant, select **Finish now, and do not start Configuration Assistant when restarting AIX** and then select **Finish**.
6. When the login prompt displays, log in as the root user to perform system administration tasks.
7. Run the **/usr/lpp/bos/post_migration** script.

Related information:

AIX Release Notes

Creating and installing a software bundle

Using this scenario, you can create a user-defined software bundle and install its contents.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

A user-defined software bundle is a text file ending in `.bnd` that is located in the `/usr/sys/inst.data/user_bundles` path. By creating the software bundle file in the `/usr/sys/inst.data/user_bundles` path, SMIT (System Management Interface Tool) can locate the file and display it in the bundle selection screen.

In this scenario, you will do the following:

- Create a user-defined software bundle that contains the `X11.apps.custom` fileset.
- Install the software bundle
- Verify the installation of the software bundle was successful

Step 1. Creating a user-defined software bundle

1. Create a text file with the extension **.bnd** in the `/usr/sys/inst.data/user_bundles` path by running the following:

   ```
   # vi /usr/sys/inst.data/user_bundles/MyBundle.bnd
   ```
2. Add the software products, packages, or filesets to the bundle file with one entry per line. Add a format-type prefix to each entry. For this example, we are dealing with AIX installp packages, so the format-type prefix is **I:**. Type the following in the *MyBundle*`.bnd` file:

   ```
   I:X11.apps.custom
   ```

   For more information on installation format types, see Software product packaging.
3. Save the software bundle file and exit the text editor.

Step 2. Installing the software bundle

1. Type the following at the command line: `# smitty easy_install`
2. Enter the name of the installation device or directory.
3. From the selection screen, select the name of the user-defined software bundle, *MyBundle*, you created.

```
                        Install Software Bundle

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.
  +-------------------------------------------------------------------------+
  |                        Select a Fileset Bundle                          |
  |                                                                         |
  | Move cursor to desired item and press Enter.                            |
  |                                                                         |
  |   App-Dev                                                               |
  |   CDE                                                                   |
  |   GNOME                                                                 |
  |   KDE                                                                   |
  |   Media-Defined                                                         |
  |   MyBundle                                                              |
  |   ...                                                                   |
  |   ...                                                                   |
  |                                                                         |
  | F1=Help                F2=Refresh             F3=Cancel                 |
  | F8=Image               F10=Exit               Enter=Do                  |
  | /=Find                 n=Find Next                                      |
  +-------------------------------------------------------------------------+
```

4. Change the values provided in the Install Software Bundle screen as appropriate to your situation. You can change the **PREVIEW only?** option to yes to preview the installation of your software bundle before you install it. You might also need to **accept new license agreements** if the software in your bundle has an electronic license.

```
                        Install Software Bundle

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

                                                      [Entry Fields]
* INPUT device / directory for software               /cdrom
* BUNDLE                                             MyBundle               +
* SOFTWARE to install                                [all]                  +
  PREVIEW only? (install operation will NOT occur)   no/yes                 +
  COMMIT software updates?                           yes                    +
  SAVE replaced files?                               no                     +
  AUTOMATICALLY install requisite software?          yes                    +
  EXTEND file systems if space needed?               yes                    +
  VERIFY install and check file sizes?               no                     +
  Include corresponding LANGUAGE filesets?           yes                    +
  DETAILED output?                                   no                     +
  Process multiple volumes?                          yes                    +
  ACCEPT new license agreements?                     no/yes                 +
  Preview new LICENSE agreements?                    no                     +

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

5. Press Enter to continue.
6. Press Enter a second time to confirm your decision and begin the installation of your software bundle.

Step 3. Verify the installation of the software bundle

- Check the installation summary at the end of the installation output by scrolling to the end of the output. The output indicates whether the installation of your user-defined software bundle was successful. You might see output similar to the following:

```
+-----------------------------------------------------------------------------+
                            Summaries:
+-----------------------------------------------------------------------------+

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
X11.apps.custom       7.2.0.0      USR      APPLY    SUCCESS
```

- You can also verify the installation at a later time by completing one of the following:
  - Run the following command:

    ```
    lslpp -Lb MyBundle
    ```

    The output indicates whether the installation of your user-defined software bundle was successful. You might see output similar to the following:

    ```
    Fileset                     Level      State  Type       Description
    ----------------------------------------------------------------------------
    X11.apps.custom  7.2.0.0               C          F      AIXwindows Customizing Tool

    State codes:
     A -- Applied.
     B -- Broken.
     C -- Committed.
     E -- EFIX Locked.
     O -- Obsolete. (partially migrated to newer version)
     ? -- Inconsistent State...Run lppchk -v.

    Type codes:

     F -- Installp Fileset
     P -- Product
     C -- Component
     T -- Feature
     R -- RPM Package
     E -- Interim Fix
    ```

  - Complete the following steps in SMIT:
    1. Type the following at a command line: `smitty list_installed`
    2. Select List Installed Software by Bundle.
    3. With your cursor at the BUNDLE name field, press F4 and select your bundle from the list.
    4. Press Enter. Output is shown similar to that in the preceding option.

Adding open source applications to your AIX system

Options for installing open source applications from the *AIX Toolbox for Linux Applications* media.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

The *AIX Toolbox for Linux Applications* media that is shipped with your base operating system software contains the most commonly used open source applications that you can use with the AIX operating system. Your options for installing from this media include:

- Using the SMIT **install_software** fast path to install **RPM** packages from the *AIX Toolbox for Linux Applications* media.
- Using the **geninstall** command to install RPM packages from the *AIX Toolbox for Linux Applications* media.
- Installing a bundle. Bundles group the applications you need for a basic Linux operating environment, basic desktop use, GNOME or KDE desktop use, or application development.

- Installing from a directory of packages classified by function. These directory groupings cover a broad range of applications, shell environments, network applications, development tools, application libraries, and so on.
- Installing a single package for a particular application.

The following procedures provide examples of installing RPM packages from *AIX Toolbox for Linux Applications* media.

- To install the **cdrecord** and **mtools** RPM packages using SMIT, do the following:
  1. Run the SMIT **install_software** fast path.
  2. Enter the device name for the *AIX Toolbox for Linux Applications* media (for example, `/dev/cd0`), and press Enter.
  3. Use the F4 key to list the contents of the device.
  4. Select the **cdrecord** and **mtools** packages, and press Enter.
  5. Accept the default values for the rest of the Install Software menu fields, and press Enter.
  6. Confirm that you do want to install the software, and press Enter.

  The software installation process begins at this point.
- To install the **cdrecord** and **mtools RPM** packages from the command line, type the following:

  ```
  # geninstall -d/dev/cd0 R:cdrecord R:mtools
  ```

  The software installation process begins at this point.
- Use the **rpm** command, which is automatically installed with the base operating system for AIX, to install the bundles required for the GNOME desktop and the **bc** application package. Complete instructions are available on the readme file for the *AIX Toolbox for Linux Applications*.
  1. With your system powered on and AIX running, insert the *AIX Toolbox for Linux Applications* media into the media drive of your system.
  2. With root authority, mount the media drive using the following command:

     ```
     mount -vcdrfs -oro /dev/cd0 /mnt
     ```

     The **-v** flag specifies the virtual file system type of **cdrfs**. The **-o** flag specifies the **ro** option, which means the mounted file is read-only. The device name is **/dev/cd0**. The directory in which you want to mount the media drive is `/mnt`.
  3. Change to the `/mnt` directory by using the following command:

     ```
     cd /mnt
     ```
  4. Use the **ls** command to list the contents of the media. The listing contains the following, which you can view or print:
     - The readme file contains complete instructions for installing from this media.
     - The CONTENTS file lists all packages available on this media and provides a short description of the purpose for each package.
  5. In your Web browser, open the `/mnt/LICENSES/index.html` file to view software licensing information.
  6. In your terminal window, change to the `ezinstall/ppc` directory by using the following command:

     ```
     cd /mnt/ezinstall/ppc
     ```

     In the next step, you use the **rpm** program to install GNOME by installing four bundles (Base, Desktop Base, GNOME Base, and GNOME Apps). Alternatively, you can install all necessary packages using the **smit install_bundle** fast path and selecting the GNOME bundle.
  7. Install GNOME by using the following sequence of commands:

```
rpm -Uhv ezinstall/ppc/base/*
rpm -Uhv ezinstall/ppc/desktop.base/*
rpm -Uhv ezinstall/ppc/gnome.base/*
rpm -Uhv ezinstall/ppc/gnome.apps/*
```

The **-U** flag updates any earlier versions of each package that you might have on your system. The **-h** flag prints hash marks (#) at timed intervals to indicate that the installation is progressing. The **-v** flag displays relevant informational or error messages that occur during the installation. Your result will look similar to the following:

```
rpm -Uhv ezinstall/ppc/desktop.base/*
gdbm            ##################################################
libjpeg         ##################################################
libpng          ##################################################
libtiff         ##################################################
libungif        ##################################################
readline        ##################################################
zlib            ##################################################
```

If your **rpm** command returns an error, it is probably caused by one of the following:

- Not enough space in your current file system. Resize the file system or change your mount point.
- Package is already installed. The **rpm** program discovered an existing package of the same name and version level, so it did not replace the package. A script on the media installs only those packages from a directory that are not already installed on your system, as shown in the following example:

  ```
  /mnt/contrib/installmissing.sh ezinstall/ppc/desktop.base/*
  ```

- Failed dependencies. The packages listed in the error message must be installed before you can install this package or bundle.

8. Install the **bc** application package by using the following command:

   ```
   rpm -Uhv RPMS/ppc/bc-*.rpm
   ```

Cloning a rootvg using alternate disk installation

Using this scenario, you can clone AIX running on **rootvg** to an alternate disk on the same system, install a user-defined software bundle, and run a user-defined script to customize the AIX image on the alternate disk.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

Because the **alternate disk installation** process involves cloning an existing **rootvg** to a target alternate disk, the target alternate disk must not be already assigned to a volume group.

In this scenario you will do the following:

- Prepare for the alternate disk installation
- Perform the alternate disk installation and customization
- Boot off the alternate disk
- Verify the operation

For information about alternate disk installation, see Installing to an alternate disk. Also, refer to the **alt_disk_copy** man page.

Step 1. Prepare for the alternate disk installation

1. Check the status of physical disks on your system. Type:

   ```
   # lspv
   ```

   Output similar to the following displays:

```
hdisk0          0009710fa9c79877    rootvg    active
hdisk1          0009710f0b90db93    None
```

We can use `hdisk1` as our alternate disk because no volume group is assigned to this physical disk.

2. Check to see if the `alt_disk_copy` fileset has been installed by running the following:

   ```
   # lslpp -L bos.alt_disk_install.rte
   ```

   Output similar to the following displays if the `alt_disk_copy` fileset is not installed:

   ```
   lslpp: 0504-132  Fileset bos.alt_disk_install.rte not installed.
   ```

3. Using volume 1 of the AIX installation media, install the **alt_disk_copy** fileset by running the following:

   ```
   # geninstall -d/dev/cd0 bos.alt_disk_install.rte
   ```

   Output similar to the following displays:

   ```
   +-----------------------------------------------------------------------------+
                                 Summaries:
   +-----------------------------------------------------------------------------+

   Installation Summary
   --------------------
   Name                        Level           Part        Event       Result
   -------------------------------------------------------------------------------
   bos.alt_disk_install.rte    5.3.0.0         USR         APPLY       SUCCESS
   ```

4. Create a user-defined bundle called **/usr/sys/inst.data/user_bundles/MyBundle.bnd** that contains the following filesets:

   ```
   I:bos.content_list
   I:bos.games
   ```

   For more information on how to create a user-defined software bundle, refer to Creating and installing a software bundle.

5. Create the **/home/scripts** directory:

   ```
   mkdir /home/scripts
   ```

6. Create a user-defined customization script called **AddUsers.sh** in the **/home/scripts** directory:

   ```
   touch /home/scripts/AddUsers.sh
   chmod 755 /home/scripts/AddUsers.sh
   ```

7. Edit **/home/scripts/AddUsers.sh** to contain the following lines:

   ```
   mkuser johndoe
   touch /home/johndoe/abc.txt
   touch /home/johndoe/xyz.txt
   ```

Step 2. Perform the alternate disk installation and customization

1. To clone the **rootvg** to an alternate disk, type the following at the command line to open the SMIT menu :

   ```
   # smit alt_clone
   ```

2. Select **hdisk1** in the **Target Disk to Install** field.
3. Select the **MyBundle** bundle in the **Bundle to Install** field.
4. Insert volume one of the installation media.
5. Type `/dev/cd0` in the **Directory or Device with images** field.
6. Type `/home/scripts/AddUsers.sh` in the **Customization script** field.
7. Press Enter to start the alternate disk installation.
8. Check that the alternate disk was created, by running the following:

   ```
   # lspv
   ```

   Output similar to the following displays:

```
hdisk0          0009710fa9c79877    rootvg
hdisk1          0009710f0b90db93    altinst_rootvg
```

Step 3. Boot from the alternate disk

1. By default, the alternate-disk-installation process changes the boot list to the alternate disk. To check this run the following:

   ```
   # bootlist -m normal -o
   ```

   Output similar to the following displays:

   ```
   hdisk1
   ```

2. Reboot the system. Type:

   ```
   # shutdown -r
   ```

   The system boots from the boot image on the alternate disk (`hdisk1`).

Step 4. Verify the operation

1. When the system reboots, it will be running off the alternate disk. To check this, type the following:

   ```
   # lspv
   ```

   Output similar to the following displays:

   ```
   hdisk0          0009710fa9c79877    old_rootvg
   hdisk1          0009710f0b90db93    rootvg
   ```

2. Verify that the customization script ran correctly, by typing the following:

   ```
   # find /home/johndoe -print
   ```

   Output similar to the following displays:

   ```
   /home/johndoe
   /home/johndoe/.profile
   /home/johndoe/abc.txt
   /home/johndoe/xyz.txt
   ```

3. Verify that the contents of your software bundle was installed, by typing the following:

   ```
   # lslpp -Lb MyBundle
   ```

   Output similar to the following displays:

   ```
     Fileset                      Level  State  Description
     ----------------------------------------------------------------------------
     bos.content_list           5.3.0.0    C    AIX Release Content List
     bos.games                  5.3.0.0    C    Games
   ```

Configuring NIM using EZNIM

Using this scenario, you will use the SMIT EZNIM option to configure the NIM environment for the first time.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

The SMIT EZNIM option installs the `bos.sysmgt.nim.master` fileset and configures the NIM environment. The configuration involves creating the NIM database and populating it with several default entries. Several basic NIM resources will then be created and defined in the NIM database.

1. Type the following: `# smitty eznim`.
2. Select **Configure as a NIM Master**, and press Enter.
3. Select **Setup the NIM Master Environment**, and press Enter.
4. Verify that the default selections for software source, volume group, and file system are correct for your environment. Change the selections, if needed.

5. Press Enter to begin configuring the NIM environment.
6. To display the NIM resources that have been created, do the following:
   a. Use the SMIT **eznim_master_panel** fast path to open the EZNIM Master menu.
   b. Select **Show the NIM environment**, and press Enter.

Installing a client using NIM

You can perform a new and complete BOS (base operating system) installation on a NIM client.

Before you install a client using NIM, you must verify that your environment meets the following configuration requirements:

- The NIM client is defined on the NIM master as a stand-alone system resource as follows.
  - The configuration is verified by running the **lsnim -l** `client_name` command.
    - If the client is not defined, then you must define it.
    - Type **smitty** `nim_mkmac` . Verify that the client is configured to be installed from the NIM master.
    - If a `/etc/niminfo` file is on the client, the **NIM_MASTER_HOSTNAME** value, is the NIM master name.
    - If this **NIM_MASTER_HOSTNAME** value is not the same as the NIM master to be used for the current installation, remove the `/etc/niminfo` file, and run **smitty** `niminit` to configure the Network Installation Management client fileset.
- The NIM master is configured and is defined with the basic NIM resources for the NIM client to be used for your installation.

To complete a BOS installing, complete the following steps:

1. Use a `bosinst_data` resource begin a non-prompted installation. For information about how to create a `bosinst.data` file for non-prompted installation, see Using the bosinst.data file.
2. Use a `resolv_conf` resource to configure the network `nameserver` and `domain`.
3. On the NIM master type the following command: `# smit nim_bosinst`.
4. From the SMIT interface, select the **lpp_source** resource for the BOS installation.
5. Select the **SPOT** resource for the BOS installation.
6. Select the **BOSINST_DATA to use during installation** option, and select a **bosinst_data** resource capable of creating a non-prompted BOS installation.
7. Select the **RESOLV_CONF to use for network configuration** option, and select a **resolv_conf** resource.
8. Select the **Accept New License Agreements** option, and select **Yes**. Accept the default values for the remaining menu options.
9. Press Enter to confirm and begin the NIM client installation.
10. To check the status of the NIM client installation, type: `# lsnim -l` *va09*. Output similar to the following displays:

```
va09:
   class          = machines
   type           = standalone
   default_res    = basic_res_grp
   platform       = chrp
   netboot_kernel = 64
   if1            = master_net va09 0
   cable_type1    = bnc
   Cstate         = Base Operating System installation is being performed
   prev_state     = BOS installation has been enabled
   Mstate         = in the process of booting
  info           = BOS install 7% complete : 0% of operations completed.
   boot           = boot
   bosinst_data   = bid_tty_ow
   lpp_source     = 720lpp_res
```

```
nim_script     = nim_script
resolv_conf    = master_net_conf
spot           = 720spot_res
cpuid          = 0009710F4C00
control        = master
Cstate_result  = success
```

Related information:

Creating system backups

Using the bosinst.data file

NIM Roadmap

Performing a nonprompted BOS installation

Network installation of a JS20 blade

Using this scenario, you can install AIX for the first time or overwrite an existing version of the operating system onto a JS20 blade.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

In this scenario, you will do the following:

- Gather the required TCP/IP information for your JS20 blade.
- Prepare your Network Installation Management (NIM) environment.
- Configure a NIM master.
- Create NIM installation resources.
- Define your JS20 blade as a NIM client.
- Prepare your JS20 blade for a network installation.
- Boot the JS20 blade off the network using a directed bootp or broadcast bootp method.

To perform a network install, you will need to configure a NIM master if you do not already have one configured. For instructions about how to create a NIM master, see Configuring NIM and other basic operations

Note: This procedure requires shutting down and reinstalling the base operating system. When you reinstall any operating system, schedule your downtime when it least impacts your workload to protect yourself from a possible loss of data or functionality. Before you perform a new and complete overwrite installation, ensure you have reliable backups of your data and any customized applications or volume groups. For instructions about how to create a system backup, see Creating system backups.

Performing a broadcast bootp network installation

To perform a broadcast bootp, ensure that the NIM server is on the same subnet as the JS20 blade that you are installing. During a broadcast bootp, the JS20 blade sends a broadcast bootp packet to its subnet. The NIM server receives and responds to this packet. The JS20 blade NIM client definition on your NIM master must include the MAC address of the JS20 blade's network adapter used during the installation, or the NIM server will not respond to a broadcast bootp. If your NIM master is on a different subnet, and you want to perform a broadcast bootp, then you must set up another system on the client's subnet to forward broadcast bootp packets.

To set up a system to forward broadcast bootp packets, complete the following steps:

1. Add the IP address of your NIM server to the `/etc/dhcprd.cnf` file on the system that forwards the packets. For example, if your NIM server's IP address is 192.24.24.1, add `server 192.24.24.1` to `/etc/dhcprd.cnf`
2. Run **startsrc -s dhcprd**.

This system will now forward broadcast bootp packets to your NIM server that is on a different subnet. You must also install the latest firmware version onto your client for the client to correctly handle the bootp response from the system that is forwarding broadcast bootp packets.

Step 1: Prepare your NIM server

Performing a broadcast bootp is similar to a directed bootp. The steps are the same, except you that must obtain the MAC address of the JS20 blade's network adapter that you will use to network install.

1. Obtain the MAC address from the MM Web interface by completing the following steps:
   a. Select **Monitor** > **Hardware VPD** .
   b. Scroll down to **BladeCenter Server MAC Addresses**.
   c. Find the JS20 blade that you plan to install and the MAC address that corresponds to the adapter you will use to perform the installation. Write this MAC address down.
2. Specify the MAC Address when you define the JS20 blade as a NIM client. If you are using the **smitty nim_mkmac** command, specify the MAC address on the **Network Adapter Hardware Address** SMIT screen. Do not include the colons (":") when you are specifying the MAC address. If the client is already defined, you can change the MAC address with the **smitty nim_chmac** command.
3. Set up your NIM master to install the JS20 blade, by completing the following steps:
   a. Run the **smitty nim_bosinst** command.
   b. Select the JS20 blade that is defined as your target .
   c. Select the type of install that you want to perform and select the installation resources that you want to use to install the JS20 blade.

   You can also prepare the JS20 blade to install using the **bos_inst** NIM operation on the command line. For more information on using the **bos_inst** operation, see Using the NIM bos_inst operation.
4. Power off the JS20 blade. If you do not want the JS20 blade to reboot automatically, set **Initiate reboot and installation now?** to **no** in the SMIT screen and press Enter.

Step 2: Initiate the installation from the management module

The bootp protocol allows you to install through a directed bootp or broadcast bootp request.

1. Ensure that the JS20 blade's boot list is set to install from the network in the MM Web interface by selecting **Blade Tasks** > **Configuration** and scroll down **Boot Sequence**.
2. Click the JS20 blade that you are installing and ensure the first device listed is **Network - BOOTP**. When the JS20 blade boots, it will install from the first network adapter that receives a bootp response.

   Note: You should not have a serial over LAN connection open to the JS20 blade that you are attempting to install when you power on the JS20 blade.
3. Click **save**.
4. Power on the JS20 blade from the MM Web interface by selecting **Blade Tasks** > **Power/Restart**.
5. Select the JS20 blade that you are installing and click **Power On Blade**.

If you do not have an serial over LAN connection to the JS20 blade, you can view the status of the installation by running the following command from your NIM master:

```
lsnim -l js20_nim_name
```

For example, if the JS20 blade was defined as **JS20blade1**, run the following command:

```
lsnim -l JS20blade1
```

Note: If you run the AIX **bootlist** command to set the IP parameters for a network adapter and reboot the system, the IP parameters will be stored in NVRAM. When you reboot the JS20 blade from the MM with the boot sequence set to **Network-BOOTP**, the JS20 blade attempts to use the IP parameters stored in NVRAM instead of performing a broadcast bootp. To perform a broadcast bootp, run the **bootlist** command specifying 0.0.0.0 for each IP parameter and reboot from AIX using the **shutdown -Fr** command. For example, to perform a broadcast bootp over **ent1**, run the following commands.

```
# bootlist -m normal ent1 client=0.0.0.0 bserver=0.0.0.0 gateway=0.0.0.0 hdisk0
# shutdown -Fr
```

If you are unable to log into the AIX system, then follow the instructions for performing a directed bootp via the Open Firmware prompt, but specify "0.0.0.0" for each IP address. Once the JS20 blade installs successfully, the boot IP parameters are reset to "0.0.0.0".

Performing a directed bootp network installation

A directed bootp can be used to install a JS20 blade from a NIM server and does not require the NIM server to be on the same subnet as the JS20 blade.

This option does not require that you have the MAC address of the network adapter on the JS20 blade. To perform a directed bootp, you need a serial over LAN connection to the blade so that you can specify the IP parameters to Open Firmware. Currently you must have 2 network adapters to perform a NIM installation if you are using serial over LAN. You cannot install AIX over the same adapter that is using serial over LAN.

Step 1: Prepare your NIM server

1. Create a SPOT, **lpp_source**, and any other resources that you will need at the level of AIX that you want to install on your NIM server. Your NIM server is usually the NIM master, but you can also set up a NIM client as a NIM server. For instructions on how to create NIM resources, see Configuring the NIM master and creating basic installation resources.
2. Ensure that you have the information in the following worksheet for your JS20 blade before proceeding with the installation:

Table 2. Network Configuration Information Worksheet

Network Attribute	Value
Network Interface	(For example: ent1)
Host Name	
IP Address	_____._____._____._____
Network Mask	_____._____._____._____
Name server	_____._____._____._____
Domain Name	
Gateway	_____._____._____._____

3. Define the JS20 blade as a NIM client on your NIM master by running the **smitty nim_mkmac** command on the NIM master. This command creates a client definition for your JS20 blade. You can also define the JS20 blade using the **define** NIM operation on the command line.
4. If you want to set the JS20 blade's name server and domain name after the installation, use a **resolv_conf** resource. For more information on creating a **resolv_conf** resource, see Using the nim_script resource.
5. Set up your NIM master to install the JS20 blade, by running the **smitty nim_bosinst** command. Select the JS20 blade that you defined earlier as your target. Then select the type of

install that you want to perform and select the installation resources that you want to use to install the JS20 blade. You can also prepare the JS20 blade to install using the **bos_inst** NIM operation on the command line.

Note:

a. If the JS20 blade is powered off or has never been installed, set **Initiate reboot and installation now?** to **no** and press enter in the SMIT interface.
b. If the JS20 blade is powered on and running AIX, set **Initiate reboot and installation now?** to **yes** in the SMIT interface. If you choose this option, a directed bootp is initiated by default and you can skip step 2. Before you run this command, ensure that the JS20 blade is a registered NIM client. To do this, run **smitty niminit** on the JS20 blade. Then specify the hostname of your NIM master and the interface you want to use for the installation. You can also initialize the JS20 blade using the **niminit** command on the command line.

Step 2: Specify a directed bootp from the JS20 blade

1. Open a Web interface to the MM by navigating to the IP address or hostname of the MM using a Web browser.
2. Enable serial over LAN to the JS20 blade from the MM Web interface by selecting **Blade Tasks** > **Serial Over LAN** .
3. Select the JS20 blade that you are installing and click **Enable Serial Over LAN**.
4. Power on the JS20 blade from the MM Web interface by selecting **Blade Tasks** > **Power/Restart**.
5. Select the JS20 blade that you are installing and click **Power On Blade**.
6. Open a serial over LAN connection to the JS20 blade by telnetting into the MM and running the **console** command. For example, if the JS20 blade is in slot 3, you would run the following command:

   ```
   console -T blade[3]
   ```

   The serial over LAN connection shows a series of LED numbers.
7. Press 8 on the keyboard when you see **E1F1** to go to the Open Firmware prompt.
8. Run **boot net:bootp,server_ip,,client_ip,gateway_ip** to boot from the network.
   - If you are using a **net** type boot, you would run a command similar to the following:

     ```
     boot net:bootp,192.168.2.10,,192.168.1.11,192.168.1.1
     ```
   - If you are using **ent1**, then you would run a command similar to the following:

     ```
     boot /pci@8000000f8000000/pci@0/ethernet@1,1:bootp,192.168.2.10,,192.168.1.11,192.168.1.1
     ```

     Note: You must specify the full device path name with this command. To determine the full path to your device, list the device tree by running the **ls** command at the Open Firmware prompt. This command displays output similar to the following:

     ```
     0 > ls
     000000c87f18: /ibm,serial
     000000c88840: /chosen
     000000c88a98: /packages

     ...
     000000d31488: /vdevice
     000000d327a8:   /vty@0
     000000d32f88:   /IBM,sp@4000
     000000d33f10:   /rtc@4001
     000000d34a18: /pci@8000000f8000000
     000000d384d0:   /pci@0
     000000d4bbd0:     /ethernet@1
     000000d5af50:     /ethernet@1,1
     000000d3be00:   /pci@3
     ```

```
000000d6a350:     /usb@0
000000d845f8:       /hub@1
000000d854b8:     /usb@0,1
000000d9f760:       /hub@1
000000d3f798:   /pci@1f
000000d45ed8:   /ide@4,1
000000d47b10:     /disk@0
```

The highlighted items are the path to the second ethernet adapter. You would pass this information to the **boot** command to initiate a network boot from the second ethernet adapter

9. After you run the **boot** command, then network installation begins. Output similar to the following is displayed on the serial over LAN connection:

```
BOOTP: chosen-network-type = ethernet,auto,none,auto
BOOTP: server   IP =        192.168.2.10
BOOTP: requested filename =
BOOTP: client   IP =        192.168.1.11
BOOTP: client   HW addr =   0 d 60 1e c cb
BOOTP: gateway  IP =        192.168.1.1
BOOTP: device    /pci@8000000f8000000/pci@0/ethernet@1,1
BOOTP: loc-code  U8842.P1Z.23A0984-P1-T7

BOOTP R = 1
FILE: /tftpboot/js20blade1.austin.ibm.com
Load Addr=0x0000000000004000, Max Size=0x0000000000bfc000
FINAL Packet Count = 21131
FINAL File Size = 10818623 bytes.
load-base=0x4000
real-base=0xc00000

Elapsed time since release of system processors: 2 mins 28 secs
...
```

Creating a system backup to tape

Using this scenario, you can create and verify a bootable system backup, also known as a *root volume group backup* or *mksysb image*

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

Step 1. Prepare for system backup creation

Before creating system backups, complete the following prerequisites:

- Be sure you are logged in as root user.
- If you plan to use a backup image for installing other differently configured target systems, you must create the image *before* configuring the source system, or set the RECOVER_DEVICES variable to `no` in the `bosinst.data` file. For more information about the `bosinst.data` file, refer to The bosinst.data file in *Installation and migration*.
- Consider altering passwords and network addresses if you use a backup to make master copies of a source system. Copying passwords from the source to a target system can create security problems. Also, if network addresses are copied to a target system, duplicate addresses can disrupt network communications.
- Mount all file systems you want to back up. The **mksysb** command backs up only mounted JFS and JFS2 in the **rootvg**. To mount file systems, use the **mount** command.

  Note: The **mksysb** command does not back up file systems mounted across an NFS network.
- Unmount any local directories that are mounted over another local directory.

Note: This backup procedure backs up files twice if a local directory is mounted over another local directory in the same file system. For example, if you mount `/tmp` over `/usr/tmp`, the files in the `/tmp` directory are then backed up twice. This duplication might exceed the number of files that a file system can hold, which can cause a future installation of the backup image to fail.

- Use the `/etc/exclude.rootvg` file to list files you do not want backed up.
- Make at least 40 MB of free disk space available in the `/tmp` directory. The **mksysb** command requires this working space for the duration of the backup.

  Use the **df** command, which reports in units of 512-byte blocks, to determine the free space in the `/tmp` directory. Use the **chfs** command to change the size of the file system, if necessary.

  For example, the following command adds 40 MB of disk space to the `/tmp` directory of a system with 4 MB partitions:

  ```
  # chfs -a size=+80000 /tmp
  ```
- All hardware must already be installed, including external devices, such as tape and media drives.
- The `bos.sysmgt.sysbr` fileset must be installed. The `bos.sysmgt.sysbr` fileset is automatically installed in AIX. To determine if the `bos.sysmgt.sysbr` fileset is installed on your system, type:

  ```
  # lslpp -l bos.sysmgt.sysbr
  ```

  If the **lslpp** command does not list the `bos.sysmgt.sysbr` fileset, install it before continuing with the backup procedure. Type the following:

  ```
  # installp -agqXd /dev/cd0 bos.sysmgt.sysbr
  ```

Step 2. Create a system backup to tape

1. Enter the `smit mksysb` fast path.
2. Select the tape device in the **Backup DEVICE or File** field.
3. If you want to create map files, select **yes** in the **Create Map Files?** field.

   Note: If you plan to reinstall the backup to target systems other than the source system, or if the disk configuration of the source system might change before reinstalling the backup, do not create map files.
4. To exclude certain files from the backup, select **yes** in the **Exclude Files** field.
5. Select **yes** in the **List files as they are backed up** field.
6. Select **yes** in the **Disable software packing of backup?** field, if you are running any other programs during the backup.
7. Use the default values for the rest of the menu options.
8. Press Enter to confirm and begin the system backup process.
9. The COMMAND STATUS screen displays, showing status messages while the system makes the backup image. When the backup process finishes, the **COMMAND:** field changes to **OK**.
10. To exit SMIT when the backup completes, press F10 (or Esc+0).
11. Remove the tape and label it. Write-protect the backup tape.
12. Record any backed-up root and user passwords. Remember that these passwords become active if you use the backup to either restore this system or install another system.

You have successfully created the backup of your **rootvg**. Because the system backup contains a boot image, you can use this tape to start your system if for some reason you cannot boot from hard disks.

Cloning a system using a system backup tape

With a **mksysb** image, you can clone one system image onto multiple target systems.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

The target systems might not contain the same hardware devices or adapters or be the same hardware platform as the source system.

All devices are installed by default during the base operating system (BOS) installation process. If the **Enable System Backups to install any system** selection in the Install Software menu was set to **yes**, you can create a **mksysb** image that boots and installs supported systems. Verify that your system is installed with all devices by typing the following:

```
# grep ALL_DEVICES_KERNELS /var/adm/ras/bosinst.data
```

Output similar to the following displays:

```
ALL_DEVICES_KERNELS = yes
```

Use this scenario if your system was not installed with all devices during BOS installation. Be sure to boot from the appropriate product media for your system and at the same maintenance or technology level of BOS as the installed source system on which the **mksysb** was made. For example, use BOS AIX media with a **mksysb** from a BOS AIX system. Use this how-to while installing a system backup tape to a different system.

In this scenario, perform the following steps:

1. Boot the system with the *AIX Volume 1* media in the media drive and the system backup tape in the tape device.

   Note: You can boot from a DVD and use a tape for the installation. However, during a tape boot, you cannot use the DVD drives to supply customized information.
2. Select **Start Maintenance Mode for System Recovery**.
3. Select **Install from a System Backup**.
4. Select the drive containing the backup tape, and press Enter.

The system reads the media and begins the installation.

You are then prompted for the BOS installation language, and the Welcome screen displays. Continue with the Prompted Installation, because cloning is not supported in nonprompted installations.

If you are cloning from the product media to restore a backup tape, do not remove the media from the media drive.

After the **mksysb** installation completes, the installation program automatically installs additional devices on your system, using the original product media you booted from. Information is saved in BOS installation log files. To view BOS installation log files, type `cd /var/adm/ras` and view the **devinst.log** file in this directory.

If the source system does not have the correct passwords and network information, you can make modifications on the target system now. Also, some products ship device-specific files. If your graphics adapter is different on the target system, verify that the device-specific filesets for graphics-related LPPs are installed.

Cleaning up a failed software installation

Using this scenario, you can clean up software products and service updates after an interrupted or failed installation.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

The cleanup procedure attempts to delete items that were partially installed or left in an incomplete state. This scenario applies only to the update or installation of optional software products. If your AIX BOS installation was unsuccessful, see Troubleshooting after a BOS installation.

Note: It is recommended that you first perform a system backup before installing software updates to ensure safe system recovery. For instructions on how to create a system backup, refer to Creating system backups.

The cleanup procedure attempts to revert the update to its previous state. For example, when cleaning up an update that was interrupted in the **COMMITTING** state, the cleanup procedure attempts to return the update to its **APPLIED** state.

If an update installation is interrupted, run the **lslpp -l** command to see the current state of the update. For example, if you run **lslpp -l** on an interrupted update installation, it might report the update status as **APPLYING** rather than **APPLIED**.

If the interruption occurs during the initial state of an installation, then the cleanup procedure attempts to delete the installation entirely and restore the previous version of the product (if there is one). When the previous version is restored, it becomes the active version. When the previous version cannot be restored, the software is listed by the **lslpp -l** command as **BROKEN**.

When the product is deleted or **BROKEN**, you can attempt to reinstall the software. Any product in the **BROKEN** state cannot be cleaned up; it can only be reinstalled or removed.

To initiate a cleanup procedure using SMIT:

1. Type `smit maintain_software` on the command line.
2. Select **Clean Up After Failed or Interrupted Installation** and press Enter.

To initiate a cleanup procedure from the command line:

Type `installp -C` on the command line and press Enter.

If prompted to reboot (restart) the system after running the cleanup procedure, then do so now.

If you get a message indicating that no products were found that could be cleaned up, you may have executed the cleanup procedure when it was not needed. Try your installation again.

Installing AIX using the media device to install a partition with an HMC

In this procedure, you will perform a new and complete base operating system installation on a logical partition using the partition's media device. This procedure assumes that there is an HMC attached to the managed system.

Prerequisites

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

Note: For the installation method that you choose, ensure that you follow the sequence of steps as shown. Within each procedure, you must use AIX to complete some installation steps, while other steps are completed using the HMC interface.

Before you begin this procedure, you should have already used the HMC to create a partition and partition profile for the client. Assign the SCSI bus controller attached to the media device, a network adapter, and enough disk space for the AIX operating system to the partition. Set the boot mode for this partition to be SMS mode. After you have successfully created the partition and partition profile, leave the partition in the *Ready* state. For instructions about how to create a logical partition and partition

profile, refer to the Creating logical partitions and partition profiles article in the IBM® Power Systems™ Hardware Information Center.

Step 1. Activate and install the partition (perform these steps in the HMC interface)

1. Activate the partition, as follows:
   a. Insert the *AIX 7 Volume 1* media into the media device of the managed system.
   b. In the navigation panel, open **Systems Management > Servers**, and click the system on which the logical partition is located.
   c. From the Tasks menu, select partition, click **Operations > Activate > Profile**.
   d. Select **Open a terminal window or console session** at the bottom of the menu to open a virtual terminal (vterm) window.
   e. Select **Advanced** to open the Advanced options menu.
   f. For the Boot mode, select **SMS**.
   g. Select **OK** to close the Advanced options menu.
   h. Select **OK**. A vterm window opens for the partition.
2. In the SMS menu on the vterm, do the following:
   a. Press the 5 key and press Enter to select **5. Select Boot Options**.

```
PowerPC Firmware
Version SF220_001
SMS 1.5 (c) Copyright IBM Corp. 2000, 2003  All rights reserved.
-------------------------------------------------------------------------------
Main Menu

1. Select Language
2. Setup Remote IPL (Initial Program Load)
3. Change SCSI Settings
4. Select Console
5. Select Boot Options


-------------------------------------------------------------------------------
Navigation Keys:

                X = eXit System Management Services
-------------------------------------------------------------------------------
Type the number of the menu item and press Enter or select Navigation Key: 5
```

   b. Press the 2 key and press Enter to select **2. Select Boot Devices**.
   c. Press the 1 key and press Enter to select **1. Select 1st Boot Device**.
   d. Press the 3 key and press Enter to select **3. DVD**.
   e. Select the media type that corresponds to the media device and press Enter.
   f. Select the device number that corresponds to the media device and press Enter. The media device is now the first device in the Current Boot Sequence list.
   g. Press the ESC key until you return to the Configure Boot Device Order menu.
   h. Select the device number that corresponds to the hard disk and press Enter.
   i. Press the x key to exit the SMS menu. Confirm that you want to exit SMS.
3. Boot from the *AIX Volume 1*, as follows:
   a. Select console and press Enter.
   b. Select language for BOS Installation menus, and press Enter to open the Welcome to Base Operating System Installation and Maintenance menu.
   c. Type 2 to select **Change/Show Installation Settings and Install** in the **Choice** field and press Enter.

```
                    Welcome to Base Operating System
                    Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

    1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    88  Help ?
    99  Previous Menu
>>> Choice [1]: 2
```

4. Verify or Change BOS Installation Settings, as follows:
   a. Type 1 in the **Choice** field to select the **System Settings** option.
   b. Type 1 for New and Complete Overwrite in the **Choice** field and press Enter.

      Note: The installation methods available depend on whether your disk has a previous version of AIX installed.
   c. When the Change Disk(s) screen opens, you can change the destination disk for the installation. If the default shown is correct, type 0 in the **Choice** field and press Enter. To change the destination disk, do the following:
      1) Type the number for each disk you choose in the **Choice** field and press Enter. *Do not* press Enter a final time until you have finished selecting all disks. If you must deselect a disk, type its number a second time and press Enter.
      2) When you have finished selecting the disks, type 0 in the **Choice** field and press Enter. The Installation and Settings screen opens with the selected disks listed under **System Settings**.
   d. If needed, change the primary language environment. Use the following steps to change the primary language used by this installation to select the language and cultural convention you want to use.

      Note: Changes to the primary language environment do not take effect until after the Base Operating System Installation has completed and your system is rebooted.
      1) Type 2 in the **Choice** field on the Installation and Settings screen to select the **Primary Language Environment Settings** option.
      2) Select the appropriate set of cultural convention, language, and keyboard options. Most of the options are a predefined combination, however, you can define your own combination of options.
         - To choose a predefined Primary Language Environment®, type that number in the **Choice** field and press Enter.
         - To configure your own primary language environment, do the following:
           a) Select **MORE CHOICES**.
           b) Select **Create Your Own Combination**.
           c) When the Set Primary Cultural Convention screen opens, type the number in the **Choice** field that corresponds to the cultural convention of your choice and press Enter.
           d) When the Set Primary Language screen opens, type the number in the **Choice** field that corresponds to your choice for the primary language and press Enter.
           e) When the Set Keyboard screen opens, type the number in the **Choice** field that corresponds to the keyboard attached to the system and press Enter.

e. After you have made all of your selections, verify that the selections are correct. Press Enter to confirm your selections and to begin the BOS Installation. The system automatically reboots after installation is complete.

5. Switch the partition to Normal Mode, as follows:
   a. Right-click on the partition profile to open the menu. Be sure the correct partition profile is highlighted.
   b. Select **Properties**.
   c. Select the **Settings** tab.
   d. For the Boot Mode, select `Normal`.
   e. Select **OK** to close the Properties menu.
   f. Right-click on the partition to open the menu.
   g. Select **Restart Partition**.
   h. Select **Immediate** for the Restart Options.
   i. Confirm that you want to restart the partition.
   j. When the partition has restarted, right-click on the partition to open the menu.
   k. Select **Open terminal window** to open a virtual terminal (vterm) window.
6. Complete the BOS Installation, as follows:
   a. Type `vt100` as the terminal type.

```
                          Set Terminal Type
The terminal is not properly initialized.  Please enter a terminal type
and press Enter.  Some terminal types are not supported in
non-English languages.

     ibm3101          tvi912          vt330
     ibm3151          tvi920          vt340
     ibm3161          tvi925          wyse30
     ibm3162          tvi950          wyse50
     ibm3163          vs100           wyse60
     ibm3164          vt100           wyse100
     ibmpc            vt320           wyse350
     lft              sun

                     +----------------------Messages-----------------------
                     | If the next screen is unreadable, press Break (Ctrl-c)
    88  Help ?       | to return to this screen.
    99  Exit         |
                     |
>>> Choice []: vt100
```

   b. In the License Agreement menu, select **Accept License Agreements**.
   c. Select **yes** to ACCEPT Installed License Agreements.
   d. Press F10 (or Esc+0) to exit the License Agreement menu.
   e. In the Installation Assistant main menu, select **Set Date and Time**.

```
                          Installation Assistant

Move cursor to desired item and press Enter.

  Set Date and Time
  Set root Password
  Configure Network Communications
  Install Software Applications
  Using SMIT (information only)


F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

f. Set the correct date, time, and time zone. Press the F3 (or Esc+3) key to return to the Installation Assistant main menu.
g. Select **Set root Password**. Set a root password for the partition.
h. Select **Configure Network Communications**. Select **TCP/IP Startup**. Select from the Available Network Interfaces and press Enter. Enter the appropriate network information in the Minimum Configuration and Startup menu and press Enter. Use the F3 (or Esc+3) key to return to the Installation Assistant main menu.
i. Exit the Installation Assistant by pressing F10 (or Esc+0).
j. The vterm window displays a login prompt.

Step 2. Manage your partition (perform this step in the AIX environment)

When the installation has completed and the system has rebooted, the vterm window displays a login prompt.

At this point, you may want to perform several common system-administration procedures. The following table lists where to find information about performing these procedures.

Table 3. **Common System Administration Procedures**

Procedure	Location
Backing up and recovering system backups	"Creating and installing system backups" in *Installation and migration*
Managing users and groups	"Users, Roles, and Passwords" in *Security*
Installing software	"Optional Software Products and Service Updates" in *Installation and migration*
Installing fixes/updates	"Optional Software Products and Service Updates" in *Installation and migration*
Tuning the system for performance	"Performance tuning" in *Performance management*
Configuring printers	*Printers and printing*

Installing AIX using the media device to install a partition without an HMC

In this procedure, you will use the system's built-in media device to perform a new and complete Base Operating System Installation on the standalone system.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

This information contains procedures to install the AIX operating system. For more information on concepts and considerations involved when performing a base operating system installation of AIX, or concepts and requirements involved when using the Network Installation Manager (NIM) to install and maintain AIX, refer to *Installation and migration*.

At this point, the BOS Installation is complete, and the initial configuration of the system is complete.

Step 1. Prepare your system for installation

- There must be adequate disk space and memory available. AIX requires 4 GB of memory and 20 GB of physical disk space. For additional release information, see the *AIX 7.2 Release Notes*.
- Make sure your hardware installation is complete, including all external devices. See the documentation provided with your system unit for installation instructions.
- If your system needs to communicate with other systems and access their resources, make sure you have the information in the following worksheet before proceeding with installation:

Table 4. Network Configuration Information Worksheet

Network Attribute	Value
Network Interface	(For example: en0, et0)
Host Name	
IP Address	_____._____._____._____
Network Mask	_____._____._____._____
Nameserver	_____._____._____._____
Domain Name	
Gateway	_____._____._____._____

Step 2. Boot from the AIX product media

1. Insert the *AIX Volume 1* media into the media device.
2. Make sure all external devices attached to the system (such as DVD drives, and terminals) are turned on. Only the media drive from which you will install AIX should contain the installation media.
3. Follow whatever procedure is needed to power on the system to cause it to boot from an AIX product media. Consult your hardware documentation for instructions if necessary.

   Note: Most older MicroChannel systems require the keylock to be set in the service position before powering on the system. Some older PCI systems require you to type 5 or press the F5 key (depending on whether you have an ASCII terminal or color graphics display console) when the system beeps and begins repeating IBM on the console several seconds after being powered on. Most current PCI systems only require that you repetitively type the 5 key (regardless of what type of console you have) at these system prompts. Also, most current systems can be set to boot from alternate media before they are powered on using the service processor menu. Consult your hardware documentation for more information.
4. Select the system console when prompted by typing the key indicated by the prompt (1, 2, F1, F2, and so on).
5. Select the English language for the base operating system (BOS) Installation menus by typing a 1 in the **Choice** field. Press Enter to open the Welcome to Base Operating System Installation and Maintenance screen.
6. Type 2 to select **2 Change/Show Installation Settings and Install** in the **Choice** field and press Enter.

```
                    Welcome to Base Operating System
                    Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

    1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    5 Select Storage Adapters

    88  Help ?
    99  Previous Menu
>>> Choice [1]: 2
```

Step 3. Set and verify BOS installation settings

1. In the Installation and Settings screen, verify the installation settings are correct by checking the method of installation (new and complete overwrite), the disk or disks you want to install, the primary language environment settings, and the advanced options.

2. To change the System Settings, which includes the method of installation and disk where you want to install, type 1 in the **Choice** field and press Enter.

```
                        Installation and Settings

Either type 0 and press Enter to install with current settings, or type the
 number of the setting you want to change and press Enter.

    1  System Settings:
         Method of Installation.............New and Complete Overwrite
         Disk Where You Want to Install.....hdisk0

>>> Choice [0]: 1
```

3. Type 1 for New and Complete Overwrite in the **Choice** field and press Enter. The Change Disk(s) Where You Want to Install screen now displays.

```
                Change Disk(s) Where You Want to Install

Type one or more numbers for the disk(s) to be used for installation and press
Enter. To cancel a choice, type the corresponding number and Press Enter.
At least one bootable disk must be selected. The current choice is indicated
by >>>.

        Name      Location Code   Size(MB)  VG Status   Bootable

     1  hdisk0   04-B0-00-2,0      30720   none            Yes
     2  hdisk1   04-B0-00-5,0      30720   none            Yes
     3  hdisk2   04-B0-00-6,0      12288   none            Yes

>>>  0   Continue with choices indicated above

    66  Disks not known to Base Operating System Installation
    77  Display More Disk Information
    88  Help ?
    99  Previous Menu

>>> Choice [0]:
```

4. In the Change Disk(s) Where You Want to Install screen:
   a. Select **hdisk0** by typing a 1 in the **Choice** field and press Enter. The disk will now be selected as indicated by **>>>**. To unselect the destination disk, type the number again and press Enter.
   b. To finish selecting disks, type a 0 in the **Choice** field and press Enter. The Installation and Settings screen displays with the selected disks listed under **System Settings**.
5. Change the Primary Language Environment Settings to English (United States). Use the following steps to change the Cultural Convention, Language, and Keyboard to English.
   a. Type 2 in the **Choice** field on the Installation and Settings screen to select the **Primary Language Environment Settings** option.
   b. Type the number corresponding to English (United States) as the Cultural Convention in the **Choice** field and press Enter.
   c. Select the appropriate keyboard and language options.
6. Verify that the selections are correct in the Overwrite Installation Summary screen, as follows:

```
     Overwrite Installation Summary

Disks:  hdisk0
Cultural Convention:  en_US
Language:  en_US
Keyboard:  en_US
Graphics Software:  Yes
Desktop:  CDE
System Management Client Software:  Yes
OpenSSH Client Software:  No
OpenSSH Server Software:  No
Enable System Backups to install any system:  Yes
Selected Edition:  express

Optional Software being installed:

>>> 1   Continue with Install
    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

7. Press Enter to begin the BOS installation. The system automatically reboots after installation is complete.

Step 4. Configure the system after installation

1. On systems with a graphics display, after a new and complete overwrite installation, the Configuration Assistant opens. On systems with an ASCII display, after a new and complete overwrite installation, the Installation Assistant opens.
2. Select the **Accept Licenses** option to accept the electronic licenses for the operating system.
3. Set the date and time, set the password for the administrator (root user), and configure network communications (TCP/IP).

   Use any other options at this time. You can return to the Configuration Assistant or the Installation Assistant by typing `configassist` or `smitty assist` at the command line.
4. Select **Exit the Configuration Assistant** and select **Next**. Or, press F10 (or `ESC+0`) to exit the Installation Assistant.
5. If you are in the Configuration Assistant, select **Finish now, and do not start Configuration Assistant when restarting AIX** and select **Finish**.

Step 5. Manage your system

At this point, you may want to perform several common system-administration procedures. The following table lists where to find information about performing these procedures.

Table 5. **Common System Administration Procedures**

Procedure	Location
Backing up and recovering system backups	"Creating and installing system backups" in *Installation and migration*
Managing users and groups	"Users, Roles, and Passwords" in *Security*
Installing software	"Optional Software Products and Service Updates" in *Installation and migration*
Installing fixes/updates	"Optional Software Products and Service Updates" in *Installation and migration*
Tuning the system for performance	"Performance tuning" in *Performance management*
Configuring printers	*Printers and printing*

Configuring the AIX system after a new installation

Using the Configuration Assistant after a new and complete overwrite installation.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

- On systems with a graphics display, after a new and complete overwrite installation, the Configuration Assistant opens.
  1. Select the **Accept Licenses** option to accept the electronic licenses for the operating system.
  2. Set the date and time, set the password for the administrator (root user), and configure network communications (TCP/IP).

     Use any other options at this time. You can return to the Configuration Assistant at any time by typing **configassist** at the command line.
  3. Select **Exit the Configuration Assistant** and select **Next**.
  4. If you are in the Configuration Assistant, select **Finish now, and do not start Configuration Assistant when restarting AIX** and select **Finish**.

  At this point, the BOS Installation is complete, and the initial configuration of the system is complete.
- On systems with an ASCII display, after a new and complete overwrite installation, the Installation Assistant opens.
  1. If the Set Terminal Type menu appears, type `vt100` as the terminal type.

```
                          Set Terminal Type
The terminal is not properly initialized.  Please enter a terminal type
and press Enter.  Some terminal types are not supported in
non-English languages.

     ibm3101          tvi912          vt330
     ibm3151          tvi920          vt340
     ibm3161          tvi925          wyse30
     ibm3162          tvi950          wyse50
     ibm3163          vs100           wyse60
     ibm3164          vt100           wyse100
     ibmpc            vt320           wyse350
     lft              sun

                     +-----------------------Messages-----------------------
                     | If the next screen is unreadable, press Break (Ctrl-c)
    88  Help ?       | to return to this screen.
    99  Exit         |
                     |
>>> Choice []: vt100
```

  2. In the License Agreement menu, select **Accept License Agreements**.
  3. Select **yes** to ACCEPT Installed License Agreements.
  4. Press F10 (or Esc+0) to exit the License Agreement menu.
  5. In the Installation Assistant main menu, select **Set Date and Time**.

```
                          Installation Assistant

Move cursor to desired item and press Enter.

  Set Date and Time
  Set root Password
  Configure Network Communications
  Install Software Applications
  Using SMIT (information only)


F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

  6. Set the correct date, time, and time zone. Press the F3 (or Esc+3) key to return to the Installation Assistant main menu.
  7. Select **Set root Password**. Set a root password for the partition.

8. Select **Configure Network Communications**. Select **TCP/IP Startup**. Select from the Available Network Interfaces and press Enter. Enter the appropriate network information in the Minimum Configuration and Startup menu and press Enter. Use the F3 (or Esc+3) key to return to the Installation Assistant main menu.
9. Exit the Installation Assistant by pressing F10 (or Esc+0).
10. The vterm window displays a login prompt.

At this point, the BOS Installation is complete, and the initial configuration of the system is complete.

Manage your AIX system after installation

At this point, you may want to perform several common system-administration procedures. The following table lists where to find information about performing these procedures.

Table 6. **Common System Administration Procedures**

Procedure	Location
Backing up and recovering system backups	"Creating and Installing System Backups" in *Installation and migration*
Managing users and groups	"Users, Roles, and Passwords" in *Security*
Installing software	"Optional Software Products and Service Updates" in *Installation and migration*
Installing fixes / updates	"Optional Software Products and Service Updates" in *Installation and migration*
Tuning the system for performance	*Performance management*
Configuring printers	*Printers and printing*

Activation Engine

Activation Engine is an enablement framework used for boot-time customization of virtual images. You can find information related to customizing a running system by performing configuration steps such as, bringing up the network interface(s), creating non-default user accounts along with their permissions, and creating new file systems.

Activation Engine overview

You can find information about the Activation Engine, and the purpose of the framework.

The Activation Engine is an enablement framework used for boot-time customization of virtual images that is executed from the `/etc/inittab` file and is processed after the initial system boot. It is used to customize the configuration settings of a system by performing functions such as starting the network interface, creating non-default user accounts along with their permissions, and creating new file systems.

The Activation Engine along with the virtual image templates allows a system administrator to use a single virtual image as a source of deployment for multiple systems, that can be customized with their own parameters such as network addresses, custom file systems, and user accounts. The Activation Engine is fully expandable, which means that you can modify the default virtual image template to add custom rules, execute custom scripts, or even add new templates that are processed at boot time. By default, the Activation Engine comes with a standard template with a predefined set of rules such as network, system accounts, file systems, and is designed to allow a user to add custom rules.

The Activation Engine script is used to parse the default virtual image template file, process all rules, and execute subsequent scripts which are linked to the processed rules. The Activation Engine supports the XML format of the template, which serves as a launchpad for calling pre-defined or user-created system customization scripts, with the script parameters being hosted in the virtual image template.

Using Activation Engine

You can find information about using the Activation Engine, required inputs, and limitations.

To use the Activation Engine follow these steps:

1. Enable and configure Activation Engine on the target system. You must enable Activation Engine on the AIX system by running the **enable** command. This process adds an Activation Engine entry to the `/etc/inittab` file, that is executed when the system boots.
2. Capture a virtual image of the virtual desktop infrastructure of the target system. This is the image you use to deploy to other systems. The target system must have the Activation Engine enabled so you can customize specific parameters when the system boots. The image is captured using the VM Control tool.
3. Create virtual image templates for the systems you want to deploy the Activation Engine.
4. Place the virtual image templates and scripts on optical drives at the appropriate location of the systems that you are deploying the Activation Engine.
5. Boot the target systems using virtual desktop infrastructure.

To configure and use the Activation Engine that is performed using the Activation Engine binary, at the `/usr/sbin/ae` file, use the following usage message:

```
/usr/sbin/ae –a {enable|disable|status|check|run}
```

enable<template> - Enables the Activation Engine

disable- Disables the Activation Engine

status- Prints current status of Activation Engine

check<template> - Validates a user created template against the Activation Engine schema

run<template> - Executes the activation engine against a particular template file

Current Limitations

Activation Engine is executed from the `/etc/inittab` file, which is executed after the initial system boot. This means that any configuration performed at the initial boot such as the NIM customization scripts is overridden by the rules set in Activation Engine templates.

The customization performed using Activation Engine script on a system is limited to changing only the configuration settings of the system because the script is run during system boot. For example, you must not use the Activation Engine script and templates to install new file sets. Activation Engine is used for changing system configuration, and must not be used to modify the user space.

The Activation Engine does not perform any input validation when parsing template files. To validate, the Activation Engine requires root access, and it is the responsibility of the user to create and store the XML template files.

During the process of enabling and disabling the Activation Engine, the `/etc/inittab` file is modified. To ensure that data integrity Activation Engine creates a backup copy of the file at `/etc/inittab.old`. This backup file is NOT deleted during the cleanup process.

Required Inputs

Input to the Activation Engine is the default image template file. Activation Engine script has a default location that it uses to search for virtual image template files in the optical media. The script attempts to mount and search the available optical media until it finds the initial template file, called `ae_template.xml`. Activation Engine uses the first template image it finds, in any of the optical discs it mounts. The default template file must be located in the root directory of the disc. If it does not find the template on any optical media it exits with an error message.

Creating AE Template File

You can find information related to the virtual image template that is the input to the AE script.

The virtual image template file is the input provided to the Activation Engine script. It is an XML file, with a specific structure that must be followed for the Activation Engine to work accurately. Each template file consists of two major portions, template settings and template data.

Schema

This following is an XML schema that is used to validate the Activation Engine template files:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="template">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="settings">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="logDirectory" type="xs:string" minOccurs="0"/>
              <xs:element name="scriptsDirectory" type="xs:string"/>
              <xs:element name="extensions" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="extendedTemplate" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="rules">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="section" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="ruleSet" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:any minOccurs="0" processContents="lax" maxOccurs="unbounded"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="name" type="xs:string" use="required"/>
                  <xs:attribute name="script" type="xs:string" use="required"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

The schema can be used to validate the custom made virtual image template files. To validate, type:

```
/usr/sbin/ae
```

with -check flag and pass the template parameter.

Document Type Description

The DTD, as a schema, can be used to ensure validity of virtual image template files. The DTD to validate Activation Engine templates is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT template (settings, rules)>
<!ATTLIST template
          name CDATA #REQUIRED>
<!ELEMENT settings (logDirectory?, scriptsDirectory, extensions?)>
<!ELEMENT logDirectory EMPTY>
<!ELEMENT scriptsDirectory EMPTY>
<!ELEMENT extensions (extendedTemplate+)>
<!ELEMENT extendedTemplate EMPTY>
<!ELEMENT rules (section+)>
<!ELEMENT section (ruleSet+)>
<!ATTLIST section
          name CDATA #REQUIRED
          script CDATA #REQUIRED>
<!ELEMENT ruleSet ANY>
```

Example

An example for ae_template.xml file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<template name="Default Activation Engine template">
 <settings>
  <!-- created automatically if it doesn't exist -->
  <logDirectory>/var/adm/ras/nim/ae/</logDirectory>
  <!-- / is assumed to be / of optical media -->
  <scriptsDirectory>/ae/scripts/</scriptsDirectory>
  <extensions>
   <extendedTemplate>/ae/user_template1.xml</extendedTemplate>
   <extendedTemplate>/ae/user_template2.xml</extendedTemplate>
  </extensions>
 </settings>
 <rules>
  <section name="network" script="ae_network.sh">
   <ruleSet>
    <address>9.3.148.163</address>
    <mask>255.255.254.0</mask>
    <gateway>9.3.148.0</gateway>
    <routes>default:0:9.3.149.1</routes>
   </ruleSet>
  </section>
  <section name="accounts" script="ae_accounts.sh">
   <ruleSet>
    <username>scott</username>
    <groups>admin,sys,system</groups>
    <admin>true</admin>
    <home>/home/bear</home>
   </ruleSet>
   <ruleSet>
    <username>eric</username>
    <groups>cron,security</groups>
    <rlogin>true</rlogin>
    <home>/home/misty</home>
   </ruleSet>
```

```
  </section>
  <section name="filesystems" script="ae_filesystems.sh">
   <ruleSet>
    <mountpoint>/usr/blah</mountpoint>
    <type>jfs2</type>
    <size>3834383</size>
    <efs>yes</efs>
    <vix>no</vix>
   </ruleSet>
   <ruleSet>
    <mountpoint>/usr/bleh</mountpoint>
    <type>jfs</type>
    <size>9595999</size>
    <efs>no</efs>
    <volume_id>Bleh</volume_id>
   </ruleSet>
  </section>
 </rules>
</template>
```

Template settings

The template settings are rules-specific to a particular template file that includes the following:

- `logDirectory`: Is a directory with the script logs. Each script has a separate log file. For example, if a script was called `ae_network_extension.sh` then its log file is `ae_network_extension.log` and is placed in the directory specified by `logDirectory` rule. If the `logDirectory` does not exist when Activation Engine starts execution, run the **mkdir** command to create the directory.
- `scriptsDirectory`: Is a directory that defines the location of the scripts. The script contains information about how each rule in the template must be linked to a particular script, and the script must be run to apply the rule. The default scripts provided are `ae_network.sh`, `ae_accounts.sh` and `ae_filesystems.sh`. These scripts contains basic functionality and must be extended for more advanced uses. The root of the path specified in `scriptsDirectory` element is assumed to be the root of the mounted optical media containing the template.
- `extensions`: Is a list of all user created virtual image templates that must be processed by the Activation Engine. Specify the templates that must be processed in order and with full file path. This list is not required if there are no user extensions to the process.

Note: The parameters of the template settings are not customizable because it is interpreted by the Activation Engine.

Sections and rulesets

The rules of a virtual image template file are an important part where all system customization parameters exist. It is subdivided into sections which are categories of rules. For example, there is a separate section for network, user accounts, and file systems. Sections are abstract separators for various groupings of system parameters. They link scripts to RuleSets. Each section has a script field where the code for customizing configuration is defined and has the rules provided in the section RuleSets.

RuleSets are subdivisions of sections. It contains a group of parameters that must be passed for a single execution of the Section script. Each RuleSet implies another different execution of the script that is linked. In one section if you want to execute the script more than once, we must have more than one RuleSet in the file system section.

Creating AE scripts

You can find information related to the AE script.

Scripts

You can create custom templates and their scripts. It is implied that if you create your templates, the template has custom scripts to execute the new rules that you have created in your templates. The **scriptsDirectory** in template settings is the place to define the location of your scripts. If you want to link a particular section to a script that must be executed by AE in your section, then you must place the script in location defined by **scriptsDirectory**.

It is expected that all the created scripts to be executed by the Activation Engine, must follow a certain set of criteria. The important requirement is that the created scripts must accept the set of arguments passed to them by the Activation Engine, as defined in the RuleSet section of the template files linking to these scripts. For example, the network section of the template file contains:

```
<section name="network" script="ae_network.sh">
        <ruleset>
          <address>9.3.148.163</address>
          <mask>255.255.254.0</mask>
          <gateway>9.3.148.0</gateway>
          <routes>default:0:9.3.149.1</routes>
        </ruleset>
</section>
```

The script **ae_network.sh** is expected to accept all three arguments as defined in the included RuleSet: **address**, **mask**, and **gateway**. The script also has to provide a proper return code to the Activation Engine. It must return 0 for SUCCESS and 1 for FAILURE.

Return code of 2 is reserved for SUCCESS_WITH_WARNINGS, which informs AE the script was successful with minor warnings and AE must log it in the execution logs. The scripts are also not expected to pipe their output to any external file. Any verbose/error messages must be sent to STDOUT, or STDERR where they are piped by AE to the appropriate destination log files, as defined in the template settings section. See, Template settings, for more details.

The scripts have certain expectations from the templates created by you. Firstly, any custom template file must follow the defined structure as defined in Schema. See “Creating AE Template File” on page 35 for more information. The template must contain a Settings section and a Rules section. The Settings section might or might not be filled out. If any or all of the rules in Settings section is not filled out by the template then its parents' rules are used. (The parent template is the `ae_template.xml` template file). Note that if the **scriptsDirectory** settings are not present in the custom template file, then the AE cannot call any scripts that are not defined in the parent template.

The rules section of the XML file must be filled out and it must follow the rigid structure of rules:

```
<rules>
      <section name="SECTION_NAME" script="SCRIPT">
          <ruleset>
              <argument1>value1</argument1>
              <argument2>value2</argument2>
          </ruleset>
      </section>
</rules>
```

In the above example, **SCRIPT** is a placeholder for a custom string that names the subscript. SCRIPT can be a filename of any system executable script, as long as its location is defined in the **scriptsDirectory** element. The above script will be run as follows:

```
PATH/SCRIPT argument1=value1 argument2=value2
```

All argument and value pairs are processed and passed to the script in the way described above. PATH represents the **scriptsDirectory** path defined in the settings section. See Template settings for more details.

Creating AE template extensions

You can find information about creating template extensions for the Activation Engine.

Template extensions

The process of linking to template extensions is similar to linking new scripts. You must define extensions list that includes your custom made templates that need to be processed by AE. Activation Engine, initially processes its default template file, `ae_template.xml` and then searches for the extension settings. If the AE find the settings, it uses that list to process template files created by you. Templates created by you must match the predefined structure of AE template files described in the section, Schema. If the newly created template file does not match the exact structure required by AE, the template is not processed by the engine.

Just as with script extensions there are certain expectations from the templates created by you. Firstly, any custom template file must follow the defined structure as defined in Schema. See Creating AE Template File for more information. The template must contain a Settings section and a Rules section. The Settings section might or might not be filled out. If any or all of the rules in Settings section is not filled out by the template then its parents' rules are used. (The parent template is the `ae_template.xml` template file). Note that if the **scriptsDirectory** settings are not present in the custom template file, then the AE cannot call any scripts that are not defined in the parent template.

Installing the Base Operating System

There are multiple ways to install the AIX base operating system.

The Base Operating System (BOS) installation program first restores the run time **bos** image, then installs the appropriate filesets, depending on your selections. The installation program automatically installs required message filesets, according to the language you choose.

If you require a minimal install, change the selections for the **Graphics Software** and the **System Management Client Software** to `no` in the **More Options** menu of the BOS menus. These options are the **GRAPHICS_BUNDLE** and **SYSTEM_MGMT_CLIENT_BUNDLE** fields in a Network Install **bosinst_data** resource. The **Enable System Backups to install any system** field is set to `yes`. This field is the **ALL_DEVICES_KERNELS** field in your **bosinst_data** resource. Performing a minimal install is only applicable for **New and Complete Overwrite** or **Preservation** installation methods.

If you are reinstalling on an older system, the DVD media can only be used to boot or reinstall on 64-bit systems. To determine if your system is a 32-bit system or a 64-bit system, run the **prtconf** command with the -c flag.

AIX Base media and AIX NIM `lpp_source` created from Base media, include updates for `bos.rte*` software. These packages are at the same V.R.M.F (version.release.modification.fix) levels as the base operating system that is restored during an operating system installation. They are also present on the media for the cases where the Base media is used to upgrade a system already at the same version and release level; to a new modification or fix level. It is recommended that you use either update media (or downloaded technology levels or service packs) to do upgrades. To support upgrading a WPAR (Workload Partition) that is moving from one system to another, the root parts of these updates are restored onto the system during an operating system installation. The data is restored into `/usr/lpp/bos/<bos.rte_software_name>/V.R.M.F/inst_root` directories. A new command, **/usr/sbin/cp_bos_updates**, is called, and is also available for users to run from the command line. If a system is installed from an `lpp_source` without the `bos.rte*` updates, running **cp_bos_updates** manually is required to support upgrading WPARs. This command allows support for WPAR Mobility, and the **restwpar** to restore a WPAR to a new system.

If you are performing a BOS installation or a `mksysb` installation on a disk with multiple ports such as Multipath I/O (MPIO), the port must remain active for the duration of the install.

Note: Before applying a Technology Level (TL), you must always create a backup and plan on restoring that backup if you need to rollback to your previous level. You can also use the **alt_disk_install** or **multibos** options as a way to get back to your previous level. Since TL updates cannot be rejected you must always commit the updates.

For more information about the installation options, refer to "BOS installation options" on page 44.

The following installation methods are available on AIX:

New and Complete Overwrite
This method installs AIX 7.2 on a new machine or completely overwrites any BOS version that exists on your system.

For instructions on installing AIX 7.2 on a new machine or to completely overwrite the BOS on an existing machine, refer to "Installing new and complete BOS overwrite or preservation" on page 57.

Preservation
This method replaces an earlier version of the BOS but retains the root volume group, the user-created logical volumes, and the **/home** file system. The system file systems **/usr**, **/var**, **/tmp**, **/opt**, and **/** (root) are overwritten. Product (application) files and configuration data stored in these file systems will be lost. Information stored in other non-system file systems will be preserved.

For instructions on preserving the user-defined structure of an existing BOS, refer to "Installing new and complete BOS overwrite or preservation" on page 57.

Migration
This method upgrades from earlier versions of the AIX BOS to AIX 7.2 (see the release notes for restrictions). The migration installation method is used to upgrade from an existing version or release of AIX to a later version or release of AIX. A migration installation preserves most file systems, including the root volume group, logical volumes, and system configuration files. It overwrites the **/tmp** file system.

For instructions on migrating an existing version or release of AIX to a later version or release of AIX, refer to "Migrating AIX" on page 405.

The following table shows the differences in the installation steps among the installation methods.

Table 7. AIX BOS Installation Methods

Installation Steps	New and Complete Overwrite	Preservation	Migration
Create **rootvg**	Yes	No	No
Create file system **/**, **/usr**, **/var**	Yes	Yes	No
Create file system **/var/adm/ras/livedump**. If this file system does not exist, it is created during any method of installation.	Yes	Yes, if not present*	Yes, if not present*
Create file system **/home**	Yes	No	No
Save Configuration	No	No	Yes
Restore BOS	Yes	Yes	Yes
Install Additional Filesets	Yes	Yes	Yes
Restore Configuration	No	No	Yes
* The livedump file system is only created during preservation or migration installations if it does not exist. You can modify the file system by using a customized **bosinst.data** file with a livedump stanza.			

Note: If you perform a migration or preservation type of installation on an existing rootvg that is running a multibos instance of AIX (`bos_*` logical volume names), the multibos instance is accepted as a

rootvg, and after the installation is complete, logical volume names are changed to the original names. It applies to both preservation and migration type of installations.

Related concepts:

"Customizing your installation" on page 81
You can customize your AIX installation. Customizing an installation requires you to edit the `bosinst.data` file and use it with your installation media.

"Migrating AIX" on page 405
During a migration, the installation process determines which optional software products are installed on the existing version of the operating system. Components from previous releases are replaced by new software in AIX Version 7.2 are installed at the AIX 7.2 level.

Related tasks:

"Migrating a multibos instance of AIX" on page 414
If you previously ran the **multibos** command to create a standby BOS, and restarted the system so that the standby BOS becomes the active BOS, and then removed the new standby BOS, you are running the AIX operating system in an environment that does not have `hd5`, `hd4`, `hd2`, `hd9var`, and `hd10opt` logical volumes, but instead the `bos_hd5`, `bos_hd4`, `bos_hd2`, `bos_hd9var`, and `bos_hd10opt` logical volumes exist. Your system is still recognized as a root volume group (rootvg) during an operating system installation, and the logical volume names are changed to their original names during the migration (or preservation) installation. If you use network alternate disk migration (**nimadm** command) to perform the migration, the logical volume names are changed when you boot the `altinst_rootvg` volume group created by the **nimadm** process for the first time.

Related information:

alt_disk_install

Using BOS menus

The available choices on the BOS menu window are described.

After you select the console and language to be used for the **BOS** menus, the **Welcome to Base Operating System Installation and Maintenance** menu displays, as follows:

```
                   Welcome to Base Operating System
                           Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

>>> 1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    5 Select Storage Adapters

    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

Note: To turn on the debug mode for the BOS installation process, type 911 in the **Choice** field and press Enter. The Welcome to Base Operating System Installation and Maintenance window refreshes and the BOS installation process runs in debug mode when the installation occurs. Continue the procedure for selecting options and specifying data until the installation begins. Debug output is sent to the client's display as the installation proceeds.

If you select **Start Install Now with Default Settings**, the BOS command determines the default installation method to use based on your system's configuration. A summary window displays, similar to the following, where you can confirm the installation method and installation options:

```
                   Overwrite Installation Summary

Disks:  hdisk0
Cultural Convention:  en_US
Language:  en_US
Keyboard:  en_US
Graphics Software:  Yes
Desktop:  CDE
System Management Client Software:  Yes
OpenSSH Client Software:  No
OpenSSH Server Software:  No
Enable System Backups to install any system:  Yes
Selected Edition:  express

Optional Software being installed:




>>> 1   Continue with Install
                     +-----------------------------------------------------
    88  Help ?       |    WARNING: Base Operating System Installation will
    99  Previous Menu|    destroy or impair recovery of ALL data on the
                     |    destination disk hdisk0.
>>> Choice [1]:
```

If the selections are correct, press Enter to begin the BOS installation.

However, if you would like to change the installation method or options, go back to the **Welcome to Base Operating System Installation and Maintenance** menu.

```
                   Welcome to Base Operating System
                         Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

>>> 1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    5 Select Storage Adapters

    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

If you want to reduce the number of disks available for selection, select option 5 **Select Storage Adapters**. Next, from the menu that lists every storage adapter on the system, select one, many, or all. If you do a preservation or migration installation, and the current rootvg is on multiple disks that are attached to multiple adapters, select all the adapters on which the rootvg disks reside.

```
                  Welcome to Base Operating System
                              Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

    1 Start Install Now with Default Settings

>>> 2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    5 Select Storage Adapters

    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

To continue and make more install time selections, select choice 2, **Change/Show Installation Settings and Install**.

The **Installation and Settings** menu displays, as follows:

```
               Installation and Settings

Either type 0 and press Enter to install with current settings, or type the
number of the setting you want to change and press Enter.

    1  System Settings:
         Method of Installation.............New and Complete Overwrite
         Disk Where You Want to Install.....hdisk0

    2  Primary Language Environment Settings (AFTER Install):
         Cultural Convention................English (United States)
         Language ..........................English (United States)
         Keyboard ..........................English (United States)
         Keyboard Type......................Default
    3  Security Model.....................Default
    4  More Options  (Software install options)
    5  Select Edition.....................express
>>> 0  Install with the current settings listed above.

                      +-----------------------------------------------------
    88  Help ?        |    WARNING: Base Operating System Installation will
    99  Previous Menu |    destroy or impair recovery of ALL data on the
                      |    destination disk hdisk0.
>>> Choice [0]:
```

For more information on the BOS menu options, refer to the Help at any time by typing 88 in the **Choice** field.

Electronic license agreements

AIX ships with software license agreements that can be viewed electronically.

If a product has an electronic license agreement, it must be accepted before software installation can continue. In the case of initial BOS installation, you can view and accept or reject license agreements in a license agreement dialog after the installation has occurred, but before the system is available for use as part of Configuration Assistant (graphics consoles) or Installation Assistant (ASCII consoles).

The AIX BOS has a license agreement, but not all software packages do. When you agree to the license agreement for BOS installation, you are also accepting all license agreements for any software installed

automatically with the BOS. Some software, such as the GNOME or KDE desktops, can be optionally installed during BOS installation; the appropriate licensing information for such software is displayed separately.

If a customized **bosinst.data** file is used (usually for unattended installations, or *nonprompted installations*), the **ACCEPT_LICENSES** field in the **control_flow** stanza can be used to accept the license agreements so users are not prompted at reboot time. When performing a "push" installation using the Network Installation Management (NIM) environment, the licenses must be accepted, either from the choices made when initializing the installation or in a customized **bosinst.data** file, before the installation can continue. For more information about the `bosinst.data` file, refer to “The bosinst.data file” on page 46.

For additional software package installations, the installation cannot occur unless the appropriate license agreements are accepted. This option, as well as options to preview licenses, is offered in both the System Management Interface Tool (SMIT) interfaces. When using the **installp** command, use the **-Y** flag to accept licenses and the **-E** flag to view license agreement files on the media.

For more information about license manipulation, refer to the **inulag** command description in the *Commands Reference, Volume 3*.

BOS installation options

The available options for installing BOS are described.

The installation options are available by typing 3 to change the **Security Model** and typing 4 to view the **More Options** field in the Installation and Settings window. These options vary based on installation type (overwrite, preservation, or migration) and security options.

The following choices are available:

Trusted AIX

Before you begin: Evaluate your system's needs for Trusted AIX with Multi Level Security (MLS) before choosing this installation option.

Applies only to overwrite and preservation installations. The **Trusted AIX** option installs the MLS version of the AIX operating system. The Trusted AIX environment enables label-based security functions in AIX, including support for:

- Labeled objects: such as files, Inter-Process Communication (IPC) objects, and network packets
- Labeled printers
- Trusted network: support for Revised Interconnection Protocol Security Option (RIPSO) and Commercial Internet Protocol Security Option (CIPSO) in Internet Protocol (IP) V4 and IP V6

After you have chosen this mode of installation, you cannot go back to a regular AIX environment without doing another overwrite install of regular AIX. For more information about Trusted AIX, see Trusted AIX.

EAL4+ configuration install *(only available with Trusted AIX)*
The **EAL4+ configuration install** option installs Trusted AIX in EAL4+ configured mode. EAL4+ configured mode provides for further restrictive security as compared to the Trusted AIX installation.

Secure by Default
Applies only to overwrite installation. The **Secure by Default** option performs a minimal software installation, and removes all clear password access such as Telnet and rlogin. Secure by Default also applies the AIX Security Expert high-security settings. Secure by Default requires direct-connect access to the system, such as TTY or direct-connect display, or a secure means of remote access such as ssh or IPsec Virtual Private Network. For more information about Secure by Default or AIX Security Expert, see Security.

Desktop
The default is **CDE** for new and complete overwrite installations. If you select **NONE**, a minimal configuration is installed including X11, Java™, perl, SMIT (if **Graphics Software** is selected).

If you select **GNOME** or **KDE**, the BOS installation process prompts you for the *AIX Toolbox for Linux Applications* media. If this media is not available, you can type `q` to continue the installation without the *AIX Toolbox for Linux Applications* media. You can select additional desktops from the Install More Software menu.

Import User Volume Groups
Applies only to migration installation and preservation installation. You have the option to have user volume groups imported after the installation completes. These volume groups can be manually imported at a later time.

Graphics Software
Applies only to new and complete overwrite installation, as well as preservation installation. Install graphics software support.

System Management Client Software
Installs Java, service agent software, and Power Systems server Console runtime software.

OpenSSH Client Software
Applies only to new and complete overwrite installation, as well as preservation installation. To install OpenSSH client software, change the choice to `Yes`.

OpenSSH Server Software
Applies only to new and complete overwrite installation, as well as preservation installation. To install OpenSSH server software, change the choice to `Yes`.

Remove Java Version 5 Software
Applies only to migration installation. By default, removes Java 5 software. You can change this choice to `No` to keep all Java Version 5 software.

Enable System Backups
If you select **Enable System Backups** to install any system, all devices are installed, so that a system backup can be installed on a different system. For more information about installing a system backup to a different system, see Cloning a system backup.

Install More Software
Applies to new and complete overwrite installation method, as well as the preservation installation method. Select **Install More Software** to choose additional software to install after the BOS installation process finishes. A software bundle file corresponds to each selection that contains the required packages and filesets. The following software bundles are available:

```
                    Install More Software

 1. Kerberos_5 (Expansion Pack)..................................... No
 2. Server  (Volume 2).............................................. No
 3. GNOME Desktop (Toolbox for Linux Applications).................. No
 4. KDE Desktop (Toolbox for Linux Applications).................... No


>>> 0  Install with the current settings listed above.

    88  Help ?
    99  Previous Menu

>>> Choice [0]:
```

The new and complete overwrite installation options (with no security models) are similar to the following:

```
                    Install Options

 1.  Desktop..................................................... NONE, CDE, KDE, GNOME
 2.  Graphics Software.............................................................. Yes
 3.  System Management Client Software....................... Yes
 4.  OpenSSH Client Software.......................................... No
 5.  OpenSSH Server Software.......................................... No
 6.  Enable System Backups to install any system........... Yes
     (Installs all devices)

>>> 7.  Install More Software

    0  Install with the current settings listed above.

    88  Help ?
    99  Previous Menu

>>> Choice [7]:
```

The migration installation options are the following:

```
                    Install Options

 1.  Enable System Backups to install any system..................... Yes
     (Installs all devices)
 2.  Import User Volume Groups....................................... Yes
 3.  Remove Java Version 5 Software.................................. Yes


>>> 0  Install with the current settings listed above.

    88  Help ?
    99  Previous Menu

>>> Choice [0]:
```

Select Edition

Type the number of this menu item to toggle through the choices of **express**, **standard**, or **enterprise**. The edition selection defines the signature file copied to the **/usr/lpp/bos** directory. The signature file is used by the **IBM License Metric Tool** (**ILMT**), to facilitate licensing compliance.

The bosinst.data file

The content and use of the `bosinst.data` file is described.

Related concepts:

"Customizing your installation" on page 81
You can customize your AIX installation. Customizing an installation requires you to edit the `bosinst.data` file and use it with your installation media.

bosinst.data file stanza descriptions

Stanza descriptions with example files are shown.

bosinst.data control_flow stanza descriptions:

The control_flow stanza contains variables that control the way the installation program works.

Variable	Description
CONSOLE	Specifies the full path name of the device you want to use as the console. If this value is **Default**, and you are performing a non-prompted installation, then the console is set to **/dev/lft0**, if this device exists. If **/dev/lft0** does not exist, the console is set to **/dev/vty0** or **/dev/tty0** depending on the system. (Instructions for which key to press are displayed on the screen.) If you change the **PROMPT** variable to **no**, you must specify a console here.
INSTALL_METHOD	Specifies a method of installation: **migrate**, **preserve**, **erase_only**, or **overwrite** (for a new and complete install). The default value is initially blank. The installation program assigns a value, depending on which version of AIX was previously installed. See "Installing the Base Operating System" on page 39 for more information. The default method of installation is **migrate** if a previous version of the operating system is on the machine. If no previous version exists, the default method is `overwrite`. The **erase_only** value specifies to erase the hard drives only and not to do an installation.
INSTALL_EDITION	Specifies the edition selection, which defines the license manager signature file to be copied to the **/usr/lpp/bos** directory on the system. The choices are **express**, **standard**, or **enterprise**, and the default option is **express**. The signature file is used by the IBM License Metric Tool (ILMT), to facilitate licensing compliance.
PROMPT	Specifies whether the installation program uses menus from which you make choices. The possible values are **yes** (default) and **no**. **Notes:** • You must fill in values for all variables in the locale stanza to uniquely identify the disk, if you set the **PROMPT** variable to **no**. Similarly, if **PROMPT** equals **no**, you must supply values for variables in the control_flow stanza, with two exceptions: the **ERROR_EXIT** and **CUSTOMIZATION_FILE** variables, which are optional. • For non-prompted installs, you must verify that the intended disks do not have reservations before you start the installation. You must use the **devrsrv** command to query the status of the disks. **Attention:** Fill in values for enough variables in the target_disk_data stanza if you set the **PROMPT** variable to **no**. The BOS installation program assigns target disks for blank variables. You can lose data if the installation program assigns a disk where you store data.
EXISTING_SYSTEM_OVERWRITE	Confirms that the installation program can *overwrite* existing volume groups. This variable is applicable only for a non-prompted overwrite installation. The possible values are **no** (default), **yes**, and **any**. **no** (Default) Only disks that are not part of a volume group can be used for the installation. **yes** Disks that contain the root volume group is used first, and if additional disks are needed for the installation, then disks that contain no volume groups are used. **any** Any disks can be used for the installation. When the installation is non-prompted and the target_disk_data stanza is empty, the installation process uses the value of the **EXISTING_SYSTEM_OVERWRITE** field to determine the disks to install on. When you do a prompted installation, this value is changed to **yes**, and is saved with other changes in the `/var/adm/ras/bosinst.data` file. Network Install Manager (NIM) creates a default `bosinst.data` file (NIM **bosinst_data** resource) with this value set to **yes**, and system backups use the `bosinst.data` file that is copied from the `/var/adm/ras` directory, so in most cases this value is already be set to **yes**. If this field is set to **no**, as seen in the `/usr/lpp/bosinst/bosinst.template` file, an error message informs you that there are not enough disks matching the criteria needed to complete the installation during a non-prompted install. The BOS installation is then changed to a prompted BOS installation, and the value of the **EXISTING_SYSTEM_OVERWRITE** field is set to **yes**.

Variable	Description
INSTALL_X_IF_ADAPTER	Specifies whether a desktop should be installed or not. The possible values are **yes**, **all**, and **no**. The default value for this field is **yes**, meaning that if the system has a graphical console, and a **DESKTOP** is specified, the desktop is installed. If set to **all** and a **DESKTOP** is specified, the desktop is installed, whether the system has a graphical console or not. If set to **no** and a **DESKTOP** is specified, the desktop is not installed.
RUN_STARTUP	Starts the Configuration Assistant on first boot after the BOS installation completes, if the system has a graphical interface. Starts Installation Assistant if the machine has an ASCII interface. The possible values are **yes** (default) and **no**. The **no** value is valid only when the **ACCEPT_LICENSES** field is set to **yes**.
RM_INST_ROOTS	Removes all files and directories in the `/usr/lpp/*/inst_roots` directories. The possible values are **no** (default) and **yes**. The `/usr/lpp/bos/inst_roots` directories must remain if the machine is used as a network server or to create workload partitions. To save disk space, set this value to **yes** if the machine is not a workload partition or network server.
ERROR_EXIT	Starts an executable program if an error occurs in the installation program. The default value is blank, which signals BOS installation to use a command that is shipped on the installation media. The command starts an error message routine when the installation program halts because of an error. As an alternative to the default, you can enter the path name of your own script or command for a customized error routine.
CUSTOMIZATION_FILE	Specifies the path name of a customization file you create. The default value is blank. The customization file is a script that starts immediately after the installation program concludes.
INSTALL_TYPE	If set to **CC_EVAL**, then CAPP and EAL4+ technology will be enabled. This is only allowed when **INSTALL_METHOD** is overwrite. If this is set, the **CULTURAL_CONVENTION** and **MESSAGES** fields of the locale stanza can only be **en_US** or **C**. Set **ALL_DEVICES_KERNELS** to **no**, and **TCB** to **yes**. **DESKTOP** can only be **NONE** or **CDE**. Set the additional software bundles to **no** (**HTTP_SERVER_BUNDLE**, **KERBEROS_5_BUNDLE**, **SERVER_BUNDLE** and **ALT_DISK_INSTALL_BUNDLE**).

BUNDLES	Specifies what software bundles to install. Type the full path name of each bundle file. Be sure there is sufficient disk space and paging space on the target machine for the software you specify in the **BUNDLES** variable.
	
	This list of bundle file names is limited to 139 bytes. If your list of bundle file names is longer than 139 bytes, use the **cat** command to combine the bundle files into a single custom bundle file and enter the name of your custom bundle file in this field.
	
	If you are installing from tape, to specify system-defined bundles on the product media, use the full path name of each bundle file as follows:
	`/usr/sys/inst.data/sys_bundles/BundleFileName`
	
	If you are using a `bosinst.data` diskette to define your own bundle files, specify the full path name of each bundle file as follows: `/../`*`DirectoryName`*`/`*`BundleFileName`*. For example, if you put a bundle file named `mybundle` in the `root` directory, the full path name would be `/../mybundle`.
	
	If you are using preservation installation, create bundle files before you start the installation. Create the files in `/home` and specify the full path name of each bundle file as follows:
	`/home/BundleFileName`

Variable	Description
RECOVER_DEVICES	Specifies whether to re-configure the devices. The default value is **Default**. For **mksysb** installations, the ODM configuration database is saved in the image. The device names and attributes are automatically extracted from the database, and the BOS installation program attempts to recreate the devices the same way they were on the machine the **mksysb** was created on. This is normal procedure for regular **mksysb** restores on the same system. However, for cloning (installing the **mksysb** image on another system), you may not want these devices configured this way, especially for network configuration. When the **mksysb** image is created, the CPU ID is saved. If you are reinstalling the same system, then the device information is recovered. If the **mksysb** image is used to install another system, device information is *not* recovered from the **mksysb** image. The **Default** value can be overwritten. For example, if your system had the planar replaced, or you upgraded to another system, you might want to recover devices. In these cases, you can select **yes** in the Backup Restore menu to recover devices.
BOSINST_DEBUG	Specifies whether to show debug output during BOS installation. The value **yes** sends **set -x** debug output to the screen during BOS installation. The possible values are **no** (default) and **yes**.
ACCEPT_LICENSES	Specifies whether to accept software license agreements during the BOS installation. The default is **no**. To automatically accept them, set this value to **yes**. When the software licenses agreements are not accepted during BOS installation, Configuration Assistant or Installation Assistant prompts you to view and accept them. During a BOS installation, if this value is blank, the default of **no** is assumed. For **mksysb** installations, when the **ACCEPT_LICENSES** field is **no**, the user is forced to accept the licenses again before continuing to use the system. When the **ACCEPT_LICENSES** field is set to **yes**, the licenses are automatically accepted for the user. If blank, the state of the licenses is the same as when the **mksysb** was created.
SYSTEM_MGMT_CLIENT_BUNDLE	Specifies whether to install Java, service agent software, and Power Systems Console software. The choices are **yes** and **no**.
OPENSSH_CLIENT_BUNDLE	Specifies to install the OpenSSH client software and the OpenSSL software that it requires. The choices are `Yes` and `No`. The default is `No`.
OPENSSH_SERVER_BUNDLE	Specifies to install the OpenSSH server software and the OpenSSL software that it requires. The choices are `Yes` and `No`. The default is `No`.
TRUSTED_AIX	Specifies the MLS version of the operating system to be installed. This is valid only with overwrite and preservation installations, and restricts other variables. The choices are **yes** and **no**. The **DESKTOP** value must be **NONE**, and **GRAPHICS_BUNDLE** must be **no**. All other software bundles must be **no**, except for **SYSTEM_MGMT_CLIENT_BUNDLE**. You can only turn off these options by installing a new operating system. For more information about Trusted AIX, see Trusted AIX.
TRUSTED_AIX_LSPP	Specifies the MLS version of the operating system to be installed, in LSPP/EAL4+ configured mode. This is a more restrictive mode of Trusted AIX. The choices are **yes** and **no**. For more information about Trusted AIX, see Trusted AIX.
SECURE_BY_DEFAULT	Specifies a minimal software installation and removes all clear password access such as telnet and rlogin. **SECURE_BY_DEFAULT** also applies the AIX Security Expert high security settings. This is only valid on an overwrite install. You must set **DESKTOP** to **NONE**, **GRAPHICS_BUNDLE** to **yes**, **ALL_DEVICES_KERNELS** to **no**, and **SYSTEM_MGMT_CLIENT_BUNDLE** to **no**. The choices are **yes** and **no**. For more information about AIX Security Expert, see AIX Security Expert.
DESKTOP	Specifies the desktop to be installed. The choice of available desktops are CDE (the default), NONE, GNOME, and KDE. If you choose GNOME or KDE, you will be prompted for the *AIX Toolbox for Linux Applications* CD.

Variable	Description
INSTALL_DEVICES_AND_UPDATES	When installing a **mksysb** image to a system with a different hardware configuration, boot from product media to get any missing device drivers installed. In addition, if the product media is a later level of AIX than the **mksysb**, software in the **mksysb** image will be updated. To prevent either of these additional installations from occurring, set this field to **no**. The default is **yes**.
IMPORT_USER_VGS	Specifies whether you want any user volume groups to be automatically imported after the system has been installed. The choices are **yes** and **no**.
ALL_DEVICES_KERNELS	Specifies whether to install all device filesets. The choices are **yes** and **no**. If you select **no**, your system will be installed with the devices and kernel specific to your system configuration. If you select **yes**, when you create a system backup of your system, you can use that system backup to install any system.
GRAPHICS_BUNDLE	Specifies whether to install the graphics software bundle during the BOS installation. This software bundle contains the graphics support for the Linux desktops. The choices are **yes** and **no**.
KERBEROS_5_BUNDLE	Specifies whether to install the Kerberos 5 client software bundle during the BOS installation. This software bundle installs the Kerberos 5 client software. The choices are **yes** and **no**.
SERVER_BUNDLE	Specifies whether to install the AIX server software bundle during the BOS installation. This software bundle installs additional networking software, performance tools, and accounting services software. The choices are **yes** and **no**.
ALT_DISK_INSTALL_BUNDLE	Specifies whether to install the alternate disk installation software during the BOS installation. The choices are **yes** and **no**.
REMOVE_JAVA_5	Specifies whether to remove the Java Version 5 software from the current system when you perform a migration installation. The choices are **yes** and **no**.
HARDWARE_DUMP	Creates a dump logical volume to contain firmware and hardware dump data. Dump logical volumes are only create on hardware that supports creation of firmware and hardware dump data. The choices are **yes** and **no**.
ERASE_ITERATIONS	Specifies the number of times to erase the chosen hard drives before the installation occurs. This field is only valid when the **INSTALL_METHOD** field is set to **overwrite** or **erase_only**. The choices for this field is a number from 0 to 8. If the field is set to 0 then no erasure of the hard drives will occur. The default is 0.
ERASE_PATTERNS	Specifies the patterns to write to the choosen hard drives. The value for this field is a comma separated list of the patterns to use for each erasure of the drives. A valid pattern is a hexadecimal value from 0 to ffffffff. The number of patterns specified must be equal or greater to the number of iterations specified in **ERASE_ITERATIONS**. If **ERASE_ITERATIONS** is 0 then this field is ignored. ex: If **ERASE_ITERATIONS** = 3 then a valid entry for this field could be **ERASE_PATTERNS** = 00,ff,0a0a0a0a .
ADD_CDE	Adds CDE as an additional desktop. If the **DESKTOP** field is not CDE and **ADD_CDE** is set to **yes**, the CDE desktop is installed in addition to the desktop specified by the **DESKTOP** field. The default value is **no**. If **DESKTOP** is set to **none**, this attribute is ignored.
ADD_KDE	Adds KDE as an additional desktop. If the **DESKTOP** field is not KDE and **ADD_KDE** is set to **yes**, the KDE desktop is installed in addition to the desktop specified by the **DESKTOP** field. The default value is **no**. If **DESKTOP** is set to **none**, this attribute is ignored.
ADD_GNOME	Adds GNOME as an additional desktop. If the **DESKTOP** field is not GNOME and **ADD_GNOME** is set to **yes**, the GNOME desktop is installed in addition to the desktop specified by the **DESKTOP** field. The default value is **no**. If **DESKTOP** is set to **none**, this attribute is ignored.
MKSYSB_MIGRATION_DEVICE	When set, specifies the device to be used to restore the **mksysb** image for migration. Default is blank. Valid values are **/dev/cd***device number* for a **mksysb** image on a CD-DVD, and **/dev/rmt***device number* for a **mksysb** image on tape. For a network installation, the valid value is the word *network*.

Variable	Description

ADAPTER_SEARCH_LIST	Specifies the adapter search list for disks. The value for this variable can be used to reduce the number of disks on which to install AIX. The field consists of a space separated list of adapters and * can be used for a group of adapters, or to specify all.
	
	Acceptable values could be:
	• scsi0 scsi1
	• fr0 scsi*
	• fr* scsi25
	• *
	
	`ADAPTER_SEARCH_LIST = scsi0 scsi1`
	`ADAPTER_SEARCH_LIST = fr0 scsi*`
	
	If an adapter specified is not defined or not available, an error occurs. If a disk in the **target_disk_data** stanza is not a child of the adapter(s) that is selected, an error occurs.
	
	If the rootvg spans multiple disks and adapters, and only one of the adapter is selected, the volume group information shows blank, as not all the disks in the volume group are being selected. If you want to install to the same rootvg disks specify each adapter associated with a disk in a rootvg.

bosinst.data target_disk_data stanza:

The target_disk_data stanza contains variables for disks in the machine where the program is to install BOS.

The default **bosinst.data** file has one target_disk_data stanza, but you can add new stanzas to install BOS on multiple disks, one stanza for each disk.

Multiple target_disk_data stanzas can exist. They define the disks that are to contain the root volume group. Only one field (**PVID**, **PHYSICAL_LOCATION**, **SAN_DISKID**, **CONNECTION**, **LOCATION**, **SIZE_MB**, **HDISKNAME**) must be non-null for BOS installation to choose a disk. The order of precedence is **PVID** (Physical Volume ID), **PHYSICAL_LOCATION**, **SAN_DISKID**, then **CONNECTION** (parent attribute//connwhere attribute), then **LOCATION**, then **SIZE_MB**, and then **HDISKNAME**. The BOS installation process uses the following logic to determine how to use the target_disk_data stanza information:

- If **PVID** is set, BOS installation checks to see if a disk matches the value. If so, other attributes are ignored.
- If **PVID** is empty and **PHYSICAL_LOCATION** is set, then BOS installation checks to see if the parent and connwhere attributes (separated by "//") match a disk. If they do, other attributes are ignored.
- If either **PVID** or **PHYSICAL_LOCATION** is set, and neither value matches a disk on the target system, and no other attributes are set, an error message is generated, and a disk must be explicitly selected.
- If **PVID** and **PHYSICAL_LOCATION** are empty, and **SAN_DISKID** is set, then, for fibre channel-attached disks, BOS installation interprets the **SAN_DISKID** as a World Wide Port Name and a Logical Unit ID (separated by "//"). The World Wide Port Name (**ww_name**) and Logical Unit ID (**lun_id**) can be obtained on a running system from the **lsattr** command.

  The **SAN_DISKID** field is checked before the **CONNECTION** field.
- If the **ww_name** and **lun_id** match a disk, other attributes are ignored.
- If either **PVID** or **SAN_DISKID** is set, and neither value matches a disk on the target system, and no other attributes are set, an error message is generated and a disk must be explicitly selected.
- If **PVID** and **SAN_DISKID** are empty and **CONNECTION** is set, BOS installation verifies if the **parent** and **connwhere** attributes (separated by "//") match a disk. If this is true, other attributes are ignored.

- If **CONNECTION** is set, the value does not match a disk on the target system, and no other attributes are set, an error message is generated and a disk must be explicitly selected.
- If other attributes are specified, processing occurs as described below:
  - If **LOCATION** is set, BOS installation ignores **SIZE_MB** and **HDISKNAME**.
  - If **LOCATION** is not set and **SIZE_MB** is, BOS installation selects disks based on **SIZE_MB** and ignores **HDISKNAME**.
  - If **LOCATION** and **SIZE_MB** are both empty, BOS installation chooses the disk specified in **HDISKNAME**.
  - If all fields are empty, BOS installation chooses a disk for you.

For the **PVID**, **PHYSICAL_LOCATION**, **SAN_DISKID**, and **CONNECTION** fields, the BOS installation process uses the following logic to determine how to use the target_disk_data stanza information:

- Does the information in one or more of the **PVID**, **PHYSICAL_LOCATION**, **SAN_DISKID**, and **CONNECTION** fields match the disk information?
- If the disk information matches the information in one of these four fields, use that information.
- If the disk information does *not* match the information in one of these four fields, and if the **LOCATION**, **SIZE_MB**, and **HDISKNAME** fields are not set, display an error message and prompt the user for the correct disk information.

The **PHYSICAL_LOCATION** information can be retrieved using the **lsdev** command. For example:

```
# lsdev -Cc disk -l hdisk0 -F "name physloc"
```

returns the `hdisk0` diskname and the `P2/Z1-A8` physical location.

> **Attention:** If **prompt=no**, do not leave the target_disk_data stanzas empty, unless it is unimportant which disk BOS installation overwrites. This is because the algorithm that determines the default disk for the installation is not always predictable.

The **SIZE_MB** field can contain either a size or the word `largest`. If a size is listed, BOS installation does a "best-fit" on the disks. If the word `largest` is in that field, BOS installation selects the largest disk. If there is more than one target_disk_data stanza, BOS installation selects the two "largest" disks, and so on.

Item	Description
PVID	Specifies the 16-digit physical volume identifier for the disk.
PHYSICAL_LOCATION	The physical location code provides a way to identify fibre channel disks during BOS Install. For fibre channel disks the **PHYSICAL_LOCATION** field includes the World Wide Port Name and Lun ID that are included in the **SAN_DISKID** field. The information in the **PHYSICAL_LOCATION** field supercedes the information in the **SAN_DISKID** field.
SAN_DISKID	Specifies the World Wide Port Name and a Logical Unit ID for fibre channel-attached disks. The **ww_name** and **lun_id** are separated by two slashes (//). This information can be obtained on a running system from the **lsattr** command.
CONNECTION	Specifies the combination of the **parent** attribute and the **connwhere** attribute associated with a disk. The parent and connwhere values are separated by two slashes (//). If the **parent** value is `scsi0` and the **connwhere** value is `0,1`, then the **CONNECTION** value is `scsi0//0,1`. This information can be obtained on a running system from the **lsdev** command. For example the **disk name**, **parent**, and **connwhere** values for all disks can be obtained by entering the following command: `lsdev -Cc disk -F "name parent connwhere"`
SIZE_MB	Specifies the formatted size of the disk, in megabytes, where the program is to install BOS. The default value is blank. You can specify the size of your target disk by typing the number of megabytes available on the formatted disk. Also, you can type `largest` if you want to use the largest disk (that has not already been selected) found by the installation program.
LOCATION	Specifies a location code for the disk where the program is to install BOS. The default value is blank. If you do not specify a value, the installation program assigns a value based on the next two variables. For more information about physical location codes, refer to the *Diagnostic Information for Multiple Bus Systems* guide.

Item	Description
HDISKNAME	Specifies the path name of the target disk. The default value is blank. To name a target disk, use the *hdiskname* format, where *hdiskname* is the device name of your disk (for example, **hdisk0**).

bosinst.data target_iscsi_data stanza:

The optional target_iscsi_data stanza contains variables for the parent iSCSI adapter of the disks in the system where the program resides to install the base operating system.

The `bosinst.data` file contains a target_iscsi_data stanza only if the root volume group includes an iSCSI disk. Only one target_iscsi_data stanza can exist. It defines the iSCSI target for the disks that are to contain the root volume group. The target_iscsi_data stanza must be located after all the target_disk_data stanzas to ensure correct processing.

Variable	Description
ADAPTER_NAME	Specifies the name of the iSCSI TOE adapter (for example, `ics0`) or the iSCSI software solution protocol device (for example, `ics0`) to which this iSCSI target will be configured. This is a required field.
ISCSI_GROUP	This field should be set to the **static** value.
TARGET_NAME	Specifies the iSCSI target name of the iSCSI target. The **mkiscsi** command will not do normalizing on the *TARGET_NAME*.
INITIATOR_NAME	Specifies the iSCSI initiator name of the iSCSI Initiator.
PORT_NUMBER	Specifies the TCP port number of the iSCSI target.
IP_ADDRESS	Specifies the IP address of the iSCSI target.
SW_INITIATOR<yes, no>	Specifies whether the adapter is an iSCSI software solution protocol device. If the adapter is an iSCSI software protocol device, the network interface that was configured by NIM is used to connect to the iSCSI target.
DISC_POLICY	Set the value of **odm**.
ADAPTER_IP	Specifies the IP address of the iSCSI TOE adapter when **SW_INITIATOR** is set to **no**.
ADAPTER_GW	Specifies the IP address of the gateway that is used by the iSCSI TOE adapter when **SW_INITIATOR** is set to **no**.
ADAPTER_SNM	Specifies the subnet mask that is used by the iSCSI TOE adapter when **SW_INITIATOR** is set to **no**.

The following example shows a target_iscsi_data stanza for a configuration where the adapter is a software initiator adapter:

```
target_iscsi_data:
ADAPTER_NAME = iscsi0
ISCSI_GROUP = static
TARGET_NAME = iqn.sn1234.iscsi_hw1
INITIATOR_NAME= iqn.2000-01.ibm.boot
PORT_NUMBER = 3260
IP_ADDRESS = 10.1.1.130
SW_INITIATOR = yes
DISC_POLICY = odm
```

The following example shows a target_iscsi_data stanza using an iSCSI TOE adapter:

```
target_iscsi_data:
ADAPTER_NAME = ics0
ISCSI_GROUP = static
TARGET_NAME = iqn.sn1234.iscsi_hw1
INITIATOR_NAME= iqn.2000-01.ibm.boot
PORT_NUMBER = 3260
IP_ADDRESS = 10.1.1.130
SW_INITIATOR = no
```

```
DISC_POLICY = odm
ADAPTER_IP = 10.1.2.115
ADAPTER_GW = 10.1.2.1
ADAPTER_SNM = 255.255.255.0
```

bosinst.data file locale stanza:

The locale stanza contains variables for the primary language that the installed machine is to use.

Refer to Understanding Locale Categories in the *AIX Globalization*, which provides information about locales and the format to use when you edit variables.

Item	Description
BOSINST_LANG	Specifies the language that the installation program uses for prompts, menus, and error messages. The default value is blank.
CULTURAL_CONVENTION	Specifies the primary locale to install. The default value is blank.
MESSAGES	Specifies the locale for message catalogs to install. The default value is blank.
KEYBOARD	Specifies the keyboard map to install. The default value is blank.

When a system backup is created and reinstalled, the default locale values are used in the `/bosinst.data` file, if available, and in the `/var/adm/ras/bosinst.data` file. These two files are not updated automatically when you change the locale value by using the **smit mlang** command. In this scenario, to match the locale value of the running system, you must change the stanza in the `/bosinst.data` file, if available, and in the `/var/adm/ras/bosinst.data` file.

bosinst.data large_dumplv stanza:

The optional **large_dumplv** stanza specifies characteristics used if a dedicated dump device is to be created on the systems.

A dedicated dump device is only created for systems with 4 GB or more of memory. The following characteristics are available for a dedicated large dump device:

Item	Description
DUMPDEVICE	Specifies the name of the dedicated dump device.
SIZEGB	Specifies the size of the dedicated dump device in gigabytes.

If the stanza is not present, the dedicated dump device is created when required. A dedicated dump device is created in machines with at least 4 Gigabytes of real memory during an overwrite install. By default, the name of the dedicated dump device is **lg_dumplv** and its size is determined by the following formula:

```
4>=  RAM < 12       size of dump device= 1 GB
12>= RAM < 24       size of dump device= 2 GB
24>= RAM < 48       size of dump device= 3 GB
     RAM >= 48      size of dump device= 4 GB
```

bosinst.data dump stanza:

The dump stanza specifies system dump characteristics.

Item	Description
PRIMARY	Specifies the primary dump device to be set by **sysdumpdev -P -p** *device*.
SECONDARY	Specifies the secondary dump device to be set by **sysdumpdev -P -s** *device*.
COPYDIR	Specifies the directory to which the dump is copied at system boot.
FORCECOPY	Specifies whether the system boots into menus that allow copy of the dump to external media if the copy fails.
ALWAYS_ALLOW	Specifies whether the key mode switch can be ignored when a dump is requested.

If the stanza is not present in the `bosinst.data` file, no additional dump-device handling occurs beyond what is already in place. Checking on the values of the fields is limited; if the device specified for a dump device is not valid, any error processing comes from the **sysdumpdev** command and is sent to the console and stored in the BOS installation log.

- If **FORCECOPY** is specified and no **COPYDIR** is specified, the value field of the **autocopydump** attribute from **/etc/objrepos/SWservAt** is retrieved and used for the **sysdumpdev -[d|D]** *copydir* **operation.**
- If only the **COPYDIR** is specified without **FORCECOPY** being specified, **forcecopy** defaults to `yes`. The **sysdumpdev -d** (**FORCECOPY** = no) or **sysdumpdev -D** (**FORCECOPY** = yes) is used to set the copy directory.
- If **ALWAYS_ALLOW=yes**, run **sysdumpdev -K**. Otherwise, run **sysdumpdev -k**.
- If any values other than `yes` and `no` are specified for **FORCECOPY** or **ALWAYS_ALLOW**, the default actions occur, and processing continues.
- If no value is specified for a particular dump field, no analogous **sysdumpdev** operation is performed. This leaves the system values in the appropriate state, even for a migration or system backup image installation. If a **COPYDIR** is specified but **FORCECOPY** is not specified, the value of the **forcecopydump** attribute is retrieved from the `/etc/objrepos/SWservAt` file to determine the correct form of **sysdumpdev** to invoke.

bosinst.data livedump stanza:

The optional livedump stanza allows you to customize the attributes of the livedump filesystem that is created during a BOS installation.

During a BOS installation, a livedump filesystem is created. To modify the attributes used to create this filesystem, use a customized `bosinst.data` file with a livedump stanza. The following attributes can be specified:

LD_DIR
The directory where the livedump filesystem will be mounted. If the LD_DIR attribute is not specified, the default directory is `/var/adm/ras/livedump`.

LD_SIZEMB
The livedump filesystem size in MB. If the LD_SIZEMB attribute is not specified, the default is 256.

LD_DEVICE
The logical-volume name for the filesystem. If the LD_DEVICE attribute is not specified, the default is livedump.

If the livedump stanza does not exist, or is not modified, the livedump filesystem is created with the default values.

Depending on the type of installation, the filesystem is created as follows:

- Overwrite installations always create a livedump filesystem.
- Preservation and migration installations create the file system as follows:

– If you specify attribute values in the livedump stanza, the filesystem is created with those values unless it already exists.
– If you do not specify attribute values in the livedump stanza, the filesystem is created with default values, if it does not already exist.

For more information about livedump, see Live Dump Facility in in *Kernel Extensions and Device Support Programming Concepts*

Using the bosinst.data file

The values in the `bosinst.data` file for this example are not specific to a network installation and can be applied for other types of installations, such as a **mksysb** installation.

Note: The depicted values illustrate formatting only and do not apply to your installation.

For information about the **bosinst.data** variable or values, see "bosinst.data file stanza descriptions" on page 46.

bosinst.data file nonprompted network installation:

An example of a modified `bosinst.data` file is shown that might be used in a nonprompted network installation.

```
control_flow:
    CONSOLE = Default
    INSTALL_METHOD = overwrite
    PROMPT = no
    EXISTING_SYSTEM_OVERWRITE = yes
    INSTALL_X_IF_ADAPTER = yes
    RUN_STARTUP = yes
    RM_INST_ROOTS = no
    ERROR_EXIT =
    CUSTOMIZATION_FILE =
    INSTALL_TYPE =
    BUNDLES =
    RECOVER_DEVICES = no
    BOSINST_DEBUG = no
    ACCEPT_LICENSES = yes
    DESKTOP = NONE
    INSTALL_DEVICES_AND_UPDATES = yes
    IMPORT_USER_VGS =
    ALL_DEVICES_KERNELS = yes
    GRAPHICS_BUNDLE = yes
    SYSTEM_MGMT_CLIENT_BUNDLE = yes
    OPENSSH_CLIENT_BUNDLE = no
    OPENSSH_SERVER_BUNDLE = no
    MOZILLA_BUNDLE = no
    KERBEROS_5_BUNDLE = no
    SERVER_BUNDLE = no
    REMOVE_JAVA_5 = yes
    HARDWARE_DUMP = yes
    ADD_CDE = no
    ADD_GNOME = no
    ADD_KDE = no
    ERASE_ITERATIONS = 0
    ERASE_PATTERNS =

target_disk_data:
    LOCATION =
    SIZE_MB =
    HDISKNAME =

locale:
```

```
BOSINST_LANG =
CULTURAL_CONVENTION =
MESSAGES =
KEYBOARD =
```

mksysb_migration_device

The device type or name where the mksysb image can be located when describing how to perform a **mksysb** or migration operation.

For a network installation, the **mksysb** image device must be a network resource and the value of "network" needs to be specified with this variable. For an installation from media, the mksysb image device needs to be specified using the device logical name as known to AIX (such as `/dev/cd0`, `/dev/rmt0`).

Installing new and complete BOS overwrite or preservation

Use these steps to install base operating system new and complete overwrite or preservation.

Step 1. Completing the prerequisites

Complete these prerequisites before starting the BOS installation.

Before starting the installation, complete the following prerequisites:

- There must be adequate disk space and memory available. AIX requires minimum of 4 GB of memory and 20 GB of physical disk space. For additional release information, see the *AIX Release Notes* that correspond to the level of your AIX operating system.
- All requisite hardware, including any external devices (such as DVD-ROM drives), must be physically connected. If you need further information, refer to the hardware documentation that accompanied your system.
- The installation media must be loaded in the boot device.
- The system *must be* set to boot from the device in which the installation media is loaded. Refer to the hardware documentation that accompanied your system for instructions on setting the boot device.
- Before you begin the installation, other users who have access to your system must be logged off.
- If the system you are installing is currently running, create or locate a backup of the system. For instructions on how to create a system backup, refer to "Creating system backups" on page 318.
- If your system needs to communicate with other systems and access their resources, make sure you have the information in the following worksheet before proceeding with installation:

Table 8. Network Configuration Information Worksheet

Network Attribute	Value
Network Adapter	
Host Name	
IP Address	_____._____._____._____
Network Mask	_____._____._____._____
Nameserver	_____._____._____._____
Domain Name	
Gateway	_____._____._____._____

Step 2. Preparing your system for installation

Prepare for a new and complete overwrite or preservation installation.

Do the following:

Note: Preservation install is only supported when moving to a newer level of the AIX Base Operating system. To re-install a prior level of AIX, you must either do a new and complete overwrite install, or re-install from a prior level system backup.

1. Insert the *AIX Volume 1* media into the media device.
2. Shut down your system. If your machine is currently running, power it off by following these steps:
   a. Log in as the root user.
   b. Type the following command:
      ```
      shutdown -F
      ```
   c. If your system does not automatically power off, place the power switch in the Off (`0`) position.
      Attention: You *must not* turn on the system unit until instructed to do so in "Step 4. Booting from your installation media" on page 59.
3. Turn on all attached external devices. These include the following:
   - Terminals
   - DVD-ROM drives
   - Monitors
   - External disk drives

   Turning on the external devices first is necessary so the system unit can identify each peripheral device during the startup (boot) process.

Step 3. Setting up an ASCII terminal

Follow these criteria for setting the communications, keyboard, and display options on an ASCII terminal.

If you are using a graphics terminal, skip directly to "Step 4. Booting from your installation media" on page 59.

If you are using an ASCII terminal, use the criteria listed below and your terminal reference documentation to set the communications, keyboard, and display options. The following settings are typical, but your terminal might have different option names and settings than those listed here.

Note: If your terminal is an IBM 3151, 3161, or 3164, press the Ctrl+Setup keys to display the Setup Menu and follow the on-screen instructions to set these options.

Table 9. Communication Options

Option	Setting
Line Speed (baud rate)	9600
Word Length (bits per character)	8
Parity	no (none)
Number of Stop Bits	1
Interface	RS-232C (or RS-422A)
Line Control	IPRTS

Table 10. Keyboard and Display Options

Option	Setting
Screen	normal
Row and Column	24x80
Scroll	jump
Auto LF (line feed)	off
Line Wrap	on
Forcing Insert	line (or both)
Tab	field
Operating Mode	echo
Turnaround Character	CR
Enter	return
Return	new line
New Line	CR
Send	page
Insert Character	space

Step 4. Booting from your installation media

Follow this procedure for booting from your installation media.

1. Turn the system unit power switch from Off (`0`) to On (|).
2. When the system beeps twice, press F5 on the keyboard (or `5` on an ASCII terminal). If you have a graphics display, you will see the keyboard icon on the screen when the beeps occur. If you have an ASCII terminal (also called a tty terminal), you will see the word `keyboard` when the beeps occur.

   Note: If your system does not boot using the F5 key (or the `5` key on an ASCII terminal), refer to your hardware documentation for information about how to boot your system from an AIX product media.

   The system begins booting from the installation media.
3. If you have more than one console, each might display a window that directs you to press a key to identify your system console. A different key is specified for each console displaying this window. If this window displays, press the specified key *only* on the console you want to use for the installation. (The system console is the keyboard and display device used for installation and system administration.)

   A window displays, asking you to select a language to be used during installation.
4. Select the language you prefer to use during installation.
5. When the Welcome to Base Operating System Installation and Maintenance window displays, type `2` in the **Choice** field to select **Change/Show Installation Settings and Install** and press Enter. Go to "Step 5. Verifying or changing the installation settings" for instructions on verifying or changing installation settings.

Step 5. Verifying or changing the installation settings

Follow this procedure for verifying or changing the installation settings.

Verify the default installation settings from the Installation and Settings window. If the installation and system settings are correct, type `0` in the **Choice** field and press Enter. Confirm that the selections on the installation summary window are correct, and press Enter to begin the BOS installation. Go to."Step 6. Finishing the BOS installation" on page 61

To change the installation settings, use the following procedure:

1. Select either **New and Complete Overwrite Installation** or **Preservation Installation**.

a. Type 1 in the **Choice** field to select the **System Settings** option.
b. When the Change Method of Installation window displays, type the number corresponding to wanted installation (either 1 for New and Complete Overwrite or 2 for Preservation) in the **Choice** field and press Enter.

   Note: Available installation methods depend on whether your system has a previous version of AIX installed.

   If you want to install the next maintenance or technology level of AIX, see."Preparing to install optional software products and service updates" on page 338 You can also use the SMIT **update_all** fast path or the **install_all_updates** command to update existing filesets to the next maintenance or technology level.
c. When the Change Disk(s) window displays, you can change the destination disk for the installation. If you selected the preservation or migration installation, the window lists only disks containing a **rootvg**.

   For New and Complete Overwrite, the disk name, the location code, the size of the disk, and the root volume group status is displayed for each available disk. The Bootable column indicates whether the disk is bootable.

   For Preservation, the level of the base operation system for the root volume group, the disks in the root volume group, the location code, and the size of the disk is displayed.

   Type 77 to select **Display More Disk Information** to view additional disk attributes such as physical volume identifier, device adapter connection location, or physical location code.

   If the default shown is correct, type 0 in the **Choice** field and press Enter. To change the destination disk, use the following procedure:

   1) Type the number for each disk you choose in the **Choice** field and press Enter. *Do not* press Enter a final time until all disks are selected. If you need to clear a disk, type its number a second time and press Enter.
      - If this installation is an overwrite installation, you can specify a supplemental disk by typing 66 and pressing the Enter key for the **Devices not known to Base Operating System Installation** option. This option opens a new menu that prompts for a device support media for the supplemental disk. The device-support media is only needed when the device cannot configure with the generic SCSI or bus-attached device drivers. BOS installation configures the system for the disk and then returns to the Change Disk window.
      - If this installation is an overwrite installation, you can specify to erase the disks chosen to be installed before the installation occurs by typing 55 and pressing the Enter key for the More Disk Options option. This option opens a new menu that prompts for the number of patterns to write, which is the number of times the drive is overwritten. If you choose 0 for the number of patterns to write, the disk is not erased before installation. This menu also prompts for the patterns to be used for each disk erasure. The patterns are a choice of the hexadecimal values 00,a5, 5a, or ff. For example, a pattern of 00 will write all zeros to the drive. Erasing a drive is a time consuming process and only drive types that are supported by the **diag** command can take advantage of this option (for example, erasure of IDE drives are not supported).
   2) When you select the disks, type 0 in the **Choice** field and press Enter. The Installation and Settings window displays with the selected disks listed under System Settings.

2. Change the primary language environment, if needed. Use the following steps to change the primary language used by this installation.

   Note: Changes to the primary language environment do not take effect until after BOS is installed and your system is rebooted.

   a. Type 2 in the **Choice** field on the Installation and Settings window to select the **Primary Language Environment Settings** option.

b. Select the appropriate set of cultural convention, language, and keyboard options. Most of the options are a predefined combination, however, you can define your own combination of options.
   - To select a predefined primary language environment, type that number in the **Choice** field and press Enter.
   - To configure your own primary language environment:
     1) Select **MORE CHOICES**.
     2) Page through the choices and select the **Create Your Own Combination** option.
     3) When the Set Primary Cultural Convention window displays, type the number in the **Choice** field that corresponds to the cultural convention of your choice and press Enter.
     4) When the Set Primary Language window displays, type the number in the **Choice** field that corresponds to your choice for the primary language and press Enter.
     5) When the Set Keyboard window displays, type the number in the **Choice** field that corresponds to the keyboard attached to the system and press Enter.

3. Change the installation options by typing 3 to change the **Security Model** or 4 to select **More Options** and press Enter. These options vary based on installation type (overwrite, preservation, or migration) and security choices. For more information about the installation options, see."BOS installation options" on page 44
4. Change the installation edition by typing 5 to toggle through the choices, **express**, **standard**, or **enterprise**. For more information about the installation options, see."BOS installation options" on page 44
5. Verify your selections in the installation summary window and press Enter to begin the BOS installation process.

Your system automatically reboots after installation is complete. Go to."Step 6. Finishing the BOS installation"

Step 6. Finishing the BOS installation

Follow this procedure for finishing the BOS installation.

1. The Installing Base Operating System window displays the status of your installation.

   After the base run-time environment is installed, status information displays about other software that is being installed.
2. The system automatically reboots.
3. After the system has restarted, you are prompted to configure your installation. For information on configuring your system after a BOS installation process, refer to "Configuring AIX" on page 85.

Note: If the system being installed has 4 GB or more of memory and you have performed an overwrite installation, then a dedicated dump device is created for you. If so, the device name is **/dev/lg_dumplv**, and its size is based on the following formula:

```
4>=  RAM < 12       size of dump device= 1 GB
12>= RAM < 24       size of dump device= 2 GB
24>= RAM < 48       size of dump device= 3 GB
     RAM >= 48      size of dump device= 4 GB
```

Related information

Links to information related to BOS installation are listed.

- For additional release information, see the *AIX Release Notes* that correspond to your level of AIX.
- For late-breaking information, which might include information about the configuration process and installed software, refer to the readme files.
- For information about installing optional software, refer to "Preparing to install optional software products and service updates" on page 338.

AIX relocatable installation

AIX relocatable installation is supported with the base AIX installation utilities, such as **installp**, **instfix**, **lslpp**, and **lppchk**. The use of relocation is of particular interest to applications that need to be installed within a Workload Partition because default System WPAR configurations do not include a writeable `/usr` or `/opt` file system. Application installations might need to be retargeted to locations other than the traditional `/usr` or `/opt` placement.

In addition to being able to install filesets in the default installation location, the system administrator can install relocatable packages into alternate root install locations. This enables the system administrator to:

- Install and maintain multiple installations of the same **installp** package in a single instance of the AIX operating system.
- Install and maintain multiple versions of the same **installp** package in a single instance of the AIX operating system.
- Use native **installp** tracking tools (such as **lppchk**, **lslpp**, **instfix**, and **inulag**) to verify and report installation data on all relocated installation instances.
- Attach and detach preinstalled software locations on a given system (such as application hosting).

User Specified Installation Location (USIL)

A User Specified Installation Location (USIL) is a tracked relocated installation path that is created by the system administrator. This location is tracked by the system and can be used as an alternate installation path for packages that support relocation.

Multiple instances and/or versions of the same software package can be installed on a single system by delegating each installation to a separate USIL. An existing USIL instance can be attached or detached from any given system.

Each USIL instance maintains its own set of Software Vital Product Data (SWVPD) in all three current **installp** parts:

- `<InstallRoot>/etc/objrepos`
- `<InstallRoot>/usr/lib/objrepos`
- `<InstallRoot>/usr/share/lib/objrepos`

Each USIL instance mirrors the default SWVPD structure within the relocated path.

USIL Management Commands	Description
/usr/sbin/mkusil	Creates or attaches a new USIL instance. `mkusil -R <RelocatePath> -c <Comments> [XFa]` Flags: **-a** Attach an existing installation as a USIL instance **-c** Comments to include in the USIL definition (visible with the **lsusil** command) **-R** Path to a new USIL location; must be a valid directory **-X** Automatically expands to space needed
/usr/sbin/lsusil	Lists existing USIL instance(s). `lsusil [-R *!ENTITY!*RelocatePath> | "ALL"]` Flags: **-R** Path to an existing USIL location

USIL Management Commands	Description
/usr/sbin/rmusil	Removes an existing USIL instance. `rmusil -R <RelocatePath>` Flags: **-R** Path to an existing USIL location **Note:** The **rmusil** command only removes the USIL reference in the SWVPD. The USIL installation path does not use any remote files.
/usr/sbin/chusil	Changes an attribute of an existing USIL instance. `chusil -R <RelocatePath> -c <NewComments> [X]` Flags: **-c** New comments to include in the USIL definition (visible with the **lsusil** command) **-R** Path to an existing USIL location **-X** Automatically expands to space needed

Listing all installation paths

Use the **lslpp** and **lppchk** commands to execute listing operations on all installation locations when the `-R "ALL"` syntax is used.

Attach and detach operations

You can use the attach operation to integrate an existing detached USIL path into the SWVPD.

For example, the administrator creates a "master" USIL instance with various relocatable applications installed for the purposes of application hosting. The administrator then copies or NFS mounts this USIL instance to various systems and uses the attach feature to integrate the USIL instance into the SWVPD.

The detach operation removes reference to the USIL instance.

installp licensing

A new USIL instance starts out with an empty LAG (**installp** license agreement ODM object class). Any installation of filesets or LPPs that require a license will require the license acceptance with the usual **installp** conventions. The license acceptance does not span USIL instances.

Relocatable installation utilities

To preserve code isolation, all USIL changes are isolated to separately compiled modules.

The relocated installation utilities include the following user level modules:

- `/usr/sbin/mkusil`
- `/usr/sbin/rmusil`
- `/usr/sbin/lsusil`
- `/usr/sbin/chusil`
- `/usr/sbin/inulag`
- `/usr/sbin/installp`
- `/usr/sbin/instfix`
- `/usr/bin/lppchk`
- `/usr/bin/lslpp`
- `/usr/sbin/inutoc`

Note: Each utility takes the **-R <*RelocatePath*>** flag. You must use these utilities when working with relocatable **installp** packages on AIX.

Relocatable applications packaging

The application packaging must support relocatable installation.

The following are recommended guidelines:

- A relocatable application package cannot deliver (write) inventory objects outside of its root install location.
- A relocatable application package cannot deliver (write) data using packaging customization outside of its root install location.
- The relocatable application package must contain the **RELOCATABLE** extended packaging attribute for each relocatable fileset. The fileset is the smallest installable unit that can be relocated.
- The relocatable application package cannot have requisites that are located in external relocated paths. It can have requisites to filesets installed in the default install path or in its own install path.

Relocatable requisites

A new packaging semantic indicates relocatable requisite location. A packager can specify that a given requisite should be found in the default install path or in the relocated install path.

The following are the new requisite semantics that apply:

prereq_ r
: **prereq** in relocated install path

ifreq_r
: **ifreq** in relocated install path

coreq_r
: **coreq** in relocated install path

instreq_r
: **instreq** in relocated install path

The currently defined requisites types (**prereq**, **ifreq**, **coreq**, and **instreq**) are all default requisites (requisites that apply to the default install location).

TOC changes for relocatable packages

The following is a sample of the new requisite sections in the TOC file:

```
sscp.rte.1.0.0.5.U.PRIVATE.bff 4 R S sscp {
sscp.rte 01.00.0000.0005 1 N B En_US Sscp
[
*coreq bos.games 1.1.1.1  <-- default requisite in default requisite section
*prereq bos.rte 1.1.1.1   <-- default requisite in default requisite section
%
/usr/bin 20
/etc 20
INSTWORK 72 40
%
%
%
IY99999  1 APAR text here.
%
RELOCATABLE <-- attribute tag to denote relocatable package
%
```

```
*prereq bos.rte 1.1.1.1 <-- default requisite in relocated requisite section
*coreq_r bos.games 1.1.1.1 <-- relocated requisite in relocated requisite section
]
}
```

- If the relocatable requisite section is present during a relocated installation, it is used as the requisite section for the installation.
- If the relocatable requisite section is not present during a relocated installation, the default requisite section is used. This means all requisites will be default requisites.
- A default installation (non-relocated) does not use the relocatable requisite section.

Relocatable application execution

The application design must support execution from an installation environment.

The following are requirements for relocatable application execution:

- The application must have a method to determine its root install location or function such that it has no dependency on the install location.
- The application must reference all application specific executable components relative to its root install location.
- The application must reference all application specific data components relative to its root install location or it must be designed to share the data with other application instances.
- The application should not make any persistent changes outside of its root install location.

USIL connector ODM class object

The USIL connector Object Data Manager (ODM) class object resides in the `/etc/objrepos/usilc` directory and contains data that links the default Software Vital Product Data (SWVPD) with all USIL instances.

The following is the object class that is contained in the `swvpd.cre` file:

```
/*  User Install Location Connector                                  */
/*  Connects the default install path to all relocated install paths.  */
class   usilc {
        vchar   path[1024];       /* USIL path     */
        vchar   comments[2048];   /* USIL Comments */
        long    flags;            /* USIL flags    */
        };
```

Note: The current SWVPD object classes include the following: **product**, **lpp**, **inventory**, **history**, **fix**, **vendor**, and **lag**.

Installing BOS on an iSCSI disk

With AIX, you can install the base operating system to an Internet Small Computer System Interface (iSCSI) disk.

To configure an iSCSI disk for base operating system use, you must supply several parameters before beginning the installation. Gather the following parameters:

Adapter Name
: Name of network adapter used for iSCSI. For iSCSI TOE adapters, this field is formatted `ics#`, where # is a number. For the iSCSI SW Initiator, this field is the Ethernet interface name and is formatted `en#`, where # is a number.

IP Address of Adapter
: IP address that is assigned to the adapter specified by Adapter Name.

IP Address of Gateway
: IP address of the gateway that is used by the adapter specified by Adapter Name.

Subnet Mask
Subnet mask that is assigned to the adapter specified by Adapter Name.

iSCSI Target Name
Name that is configured for the iSCSI Target.

iSCSI Initiator Name
Initiator name that is configured for the iSCSI Target.

Port Number
Port Number that is configured for the iSCSI Target.

IP Address of Target
IP Address that is configured for the iSCSI Target.

Notes:

1. Consult your iSCSI vendor's documentation for more information.
2. IPv6 support for iSCSI disk installation is not supported.
3. iSCSI boot is supported by using the iSCSI software initiator when you run POWER6® or later processors.
4. iSCSI boot is supported by using the iSCSI TOE daughtercard in POWER® processor-based blade systems. iSCSI boot is not supported by using the iSCSI TOE PCI slot adapter.
5. When you boot by using the iSCSI software initiator, ensure that the Ethernet network is configured so that the link is enabled without delay. After the Ethernet link is enabled, the AIX iSCSI software initiator attempts to contact the iSCSI target for approximately 30 seconds before declaring that the boot disk cannot be found and indicating the **554 Unknown Boot Disk** error. Some Ethernet protocols, such as spanning tree protocols, might prevent the link from being enabled in 30 seconds and might cause boot failures. Such protocols must be disabled or overridden on the Ethernet switch if they prevent the Ethernet link from being enabled in less than 30 seconds.
6. You cannot install the Base Operating System (BOS) on an iSCSI disk from a tape drive. You can install BOS on an iSCSI disk only through Network Installation Management (NIM) or DVD/CD.

For prompted installs of AIX, these parameters can be submitted using the iSCSI configuration menus. For non-prompted installs of AIX, these parameters can be supplied using the **bosinst.data file stanza descriptions**.

Only one path to one iSCSI target can be configured for the root volume group used to install the base operating system. The root volume group cannot be created by combining iSCSI disks with non-iSCSI disks. After installation is complete, additional iSCSI paths may be configured for the root volume group if MPIO access to the root volume group is required.

Note: Support for MPIO iSCSI boot is available only on Power9 systems running firmware level FW920, or later. Support for single path iSCSI boot is available on Power6 systems, or later.

Using the iSCSI configuration menus

The iSCSI configuration menus can be accessed from the " Make Additional Disks Available" choice on the main menu of the **Base Operating System** menus.

To access the iSCSI configuration menus, perform the following steps:

1. From the base operating system (BOS) menus, select **Welcome to Base Operating System Installation and Maintenance**.
2. Choose the **Make Additional Disks Available** option.
3. Choose the **Configure Network Disks (iSCSI)** option to load the iSCSI configuration menus.
4. At the **Configure iSCSI** SMIT menu, select the **iSCSI Configuration** option.
5. Enter the iSCSI parameters and press Enter.

After the menu is submitted, you can see the output from the configuration commands, as well as output listing the iSCSI disks that have successfully been configured.

6. If the correct iSCSI disks have been configured, proceed to BOS installation by pressing **F10** to exit to BOS menus.
7. Select **Change/Show Installation Settings and Install** to select the iSCSI disks for installation.

Accessing maintenance mode to recover iSCSI parameters

If you are unable to start the BOS from an iSCSI disk, you might need to access maintenance mode to reconfigure the iSCSI parameters used during boot.

Access maintenance mode using a CD or DVD boot with the installation media. For more information, see Accessing the system if unable to boot from the hard disk. Maintenance mode can also be accessed by starting the network using NIM. For more information, see Booting in maintenance mode. After you have accessed maintenance mode, perform the following steps:

1. Select **Configure Network Disks (iSCSI)**. The iSCSI configuration menus are launched.

   Note: If you supplied all of the iSCSI parameters through a `bosinst.data` file using NIM, this step might not be required.
   When the correct disk has been configured, exit the iSCSI configuration menus.
2. At the Maintenance menu, save the iSCSI configuration parameters for the disk configured in the previous step to the root volume group on the disk. Select option 1, **Access a Root Volume Group**. The Warning screen is displayed.
3. Read the information displayed on the Warning screen. When you are ready to continue, type `0` and press Enter. The Access a Root Volume Group menu is displayed.
4. Select the root volume group on the disk that was configured in Step 2. After entering your selection, the Volume Group Information menu is displayed.

   Note: Reviewing the disk and location code information on the Volume Group Information menu enables you to determine whether the volume group you selected was the root volume group. You can return to the Access a Root Volume Group screen if the choice you made was not the root volume group. If you have not chosen a root volume group, you cannot continue beyond the Volume Group Information menu.
5. Select **Choice 1** from the Volume Group Information menu and press Enter. A shell and system prompt is displayed.
6. At the system prompt, run the **update_iscsi** command to save the iSCSI configuration to the root volume group. The system can now be restarted using the updated iSCSI parameters.
7. Run the **bootlist** command, specifying the **hdisk** option being used for booting the system, as configured in step 2. For example, run **bootlist -m normal hdisk3** if **hdisk3** is the new iSCSI boot disk.

Note: The **update_iscsi** command might change the state of some network interfaces to **down** and help avoid possible conflicts with the changes made to the network interface that are used to access the iSCSI boot disk. After the system starts, examine the network interfaces. Delete any network interfaces that are no longer valid, and bring up any network interfaces that are still valid but were marked **down** by the **update_iscsi** command.

For more information, see Using the iSCSI configuration menus.

Installing BOS to an alternate disk

Alternate disk installation lets you install the operating system while it is still up and running, which reduces installation or upgrade downtime considerably.

Alternate disk installation also allows large facilities to better manage an upgrade because systems can be installed over a longer period of time. While the systems are still running at the previous version, the switch to the newer version can happen at the same time.

Alternate disk installation filesets

An alternate disk installation uses these filesets.

Item	Description
bos.alt_disk_install.boot_images	Must be installed for alternate disk **mksysb** installations.
bos.alt_disk_install.rte	Must be installed for **rootvg** cloning and alternate disk **mksysb** installations.

Installing an alternate mksysb disk

Alternate **mksysb** installation involves installing a **mksysb** image that has already been created from a system, onto an alternate disk of the target system. The alternate disk or disks cannot contain a volume group.

The **mksysb** image is created on a system that either was the same hardware configuration as the target system, or had all the device and kernel support installed for a different machine type or platform, or different devices. The installed device and kernel support would be as follows:

- **devices.***
- **bos.mp64**

Note: In AIX, all device and kernel support is automatically installed during a base operating system installation.

When the **alt_disk_mksysb** command is run, the `image.data` file from the **mksysb** image is used by default (unless a customized **image.data** is given) to create the logical volumes and file systems. The prefix **alt_** is added to the logical volume names, and the file systems are created with a prefix of **/alt_inst**. For example, `hd2` would be created as **alt_hd2**, and its file system, [1], would be created as `/alt_inst/usr`. These names are changed back to their original names at the end of the alternate disk installation process.

The **mksysb** image is then restored into the alternate file system. A prepopulated boot image is then copied to the boot logical volume of the **altinst_rootvg**, and the boot record of the boot disk is modified to allow booting from the disk.

At this point, a script can be run to allow for any customization before the system is rebooted. The alternate file systems are still mounted as `/alt_inst/`***real_file_system*** (for example: `/alt_inst/usr`, `/alt_inst/home`). Files can be accessed at this point, but nothing can be installed into the alternate file system because the kernels and libraries of the **mksysb** image may not match those of the running system.

After the optional script is run, the file systems are unmounted, and the logical volume and file system names are changed to match the `image.data` file's names (for example, `alt_inst_hd6` is changed to `hd6` in the volume group descriptor area). The logical volumes are exported from the Object Data Manager (ODM), but the **altinst_rootvg** is only varied off. It is left in the ODM as a placeholder so the disk is not accidentally overwritten. The default action of the **alt_disk_mksysb** command is to set the bootlist so that the next time the system boots, it boots from this newly installed volume group. This default action can be turned off. If specified, the system reboots at this point, and the system reboots from the new **rootvg**. The boot process proceeds to a certain point, with the new **rootvg**'s file systems mounted, and the **bosboot** command is called to rebuild a "normal" boot logical volume. The system then reboots.

1. /usr

After rebooting from the new alternate disk, the former **rootvg** volume group is contained in an **lspv** listing as `old_rootvg`, and includes all disk(s) in the original **rootvg**. This former **rootvg** volume group is set to `not varyon` at reboot and should *only* be removed with the **-X** flag. For example:

```
alt_rootvg_op -X old_rootvg
```

If a return to the original **rootvg** is necessary, the **bootlist** command is used to change the bootlist to reboot from the original **rootvg**.

If it is unclear which disk is the boot disk for a specific volume group, use the **-q** flag to determine the boot disk. This flag can be useful when a volume group comprises multiple disks and a change in the bootlist is necessary.

Cloning the rootvg to an alternate disk

Cloning the **rootvg** to an alternate disk has many advantages. One advantage is having an online backup available, in case of a disk crash. Keeping an online backup requires an extra disk or disks to be available on the system.

Another benefit of **rootvg** cloning occurs when applying new maintenance or technology level updates. A copy of the **rootvg** is made to an alternate disk, then updates are applied to that copy. The system runs uninterrupted during this time. When it is rebooted, the system boots from the newly updated **rootvg** for testing. If updates cause problems, the **old_rootvg** can be retrieved by resetting the bootlist and then rebooting.

If your current rootvg uses the JFS file system, then the alternate disk cannot have 4K sector sizes.

By default, calling the **alt_disk_install** command does the following:

1. Creates an `/image.data` file based on the current **rootvg**'s configuration. A customized `image.data` file can be used.
2. Creates an alternate **rootvg** (**altinst_rootvg**).
3. Creates logical volumes and file systems with the **alt_inst** prefix.
4. Generates a backup file list from the **rootvg**, and if an `exclude.list` file is given, those files are excluded from the list.
5. Copies the final list to the **altinst_rootvg**'s file systems.
6. If specified, the **installp** command installs updates, fixes, or new filesets into the alternate file system.
7. The **bosboot** command creates a boot logical volume on the alternate boot disk.
8. If a customization script is specified, it runs at this point.
9. The file systems are then unmounted, and the logical volumes and file systems are renamed.
10. The logical volume definitions are exported from the system to avoid confusion with identical ODM names, but the **altinst_rootvg** definition is left as an ODM placeholder.
11. By default, the bootlist is set to the new cloned **rootvg** for the next reboot.

Performing an alternate disk phased installation:

The alternate disk installation can be performed in stages.

The installation is broken down into three phases. The default is to perform all three phases in the same invocation. The phases are as follows:

Item	Description
Phase 1	Creates the **altinst_rootvg** volume group, the **alt_** logical volumes, and the `/alt_inst` file systems. Also restores the **mksysb** or **rootvg** data.
Phase 2	Runs any specified customization script. For cloning only, installs updates, new filesets, fixes, or bundles. Also copies a `resolv.conf` file (if specified) and necessary files to remain a NIM client (if specified).
Phase 3	Unmounts the `/alt_inst` file systems, renames the file systems and logical volumes, removes the **alt_** logical volume names from ODM, and varies off the **altinst_rootvg**. It also sets the bootlist and reboots (if specified).

As an alternative to running all three phases, the phases can be completed by one of the following methods:

- Each phase separately
- Phases 1 and 2 together
- Phases 2 and 3 together (Phase 2 can be run multiple times before Phase 3 is run.)

You must run Phase 3 to obtain a usable **rootvg**. Running Phases 1 and 2 leave the `/alt_inst` file systems mounted. Any time during the phase process and before rebooting, the **altinst_rootvg** can be removed, and disk cleanup occurs using the following command:

```
alt_rootvg_op -X
```

Performing an alternate disk migration installation

Alternate disk migration installation allows you to create a copy of **rootvg** to a free disk, or disks, and simultaneously migrate it through Network Installation Management (NIM) to a new release level.

Using alternate disk migration installation compared to a conventional migration provides the following advantages:

- Reduced downtime; the migration is performed while the system is up normally, and there is no need to boot from any media.
- Quick recovery in case of migration failure.
- High degree of flexibility and customization.

Reduced downtime. The migration is performed while the system is up and functioning. There is no requirement to boot from install media, and the majority of processing occurs on the NIM master.

Quick recovery in the event of migration failure. Because you are creating a copy of **rootvg**, all changes are performed to the copy (**altinst_rootvg**). In the event of serious migration installation failure, the failed migration is cleaned up, and there is no need for the administrator to take further action. In the event of a problem with the new (migrated) level of AIX, the system can be quickly returned to the premigration operating system by booting from the original disk.

High degree of flexibility and customization in the migration process. This is done with the use of optional NIM customization resources including **image_data**, **bosinst_data**, **exclude_files**, premigration script, **installp_bundle**, and post-migration script.

Network Install Manager Alternate Disk Migration (**nimadm**) is a utility that allows you to do the following:

- Create a copy of rootvg to a free disk, or disks, and simultaneously migrate it to a new version or release level of AIX.
- Using a copy of rootvg, create a new nim mksysb resource that has been migrated to a new version or release level of AIX.
- Using a **nim mksysb** resource, create a new nim **mksysb** resource that has been migrated to a new version or release level of AIX.

- Using a **nim mksysb** resource, restore to a free disk, or disks, simultaneously migrating it to a new version or release level of AIX.

nimadm uses NIM resources to perform these functions.

For more information about the **nimadm** command, refer to the *Commands Reference*.

Preparing for an alternate disk migration:

These are the requirements for an alternate disk migration installation.

1. The NIM master must have the same level of **bos.alt_disk_install.rte** installed in its **rootvg** and the **SPOT** which is used to perform the migration.

   Note: It is not necessary to install the alt_disk_install utilities on the client
2. The selected **lpp_source** NIM resource, and selected SPOT NIM resource must match the AIX level to which you are migrating.
3. The NIM master must be at the same or higher AIX level then the level being migrated to.
4. The client or the system to be migrated, must be at a prior version or release of AIX, than the level being migrated to.
5. The client must have a disk large enough to clone the **rootvg** and approximately an additional 500 Megs of free space for the migration. The total amount of required space depends on original system configuration and **nimadm** customization.
6. The target client must be a registered with the master as a stand alone NIM client.
7. Beginning with AIX 61TL 8 and AIX 71 TL2, the NIM client can be configured to communicated with the NIM master using **NIMSH** for alternate disk migration. The NIM master must be able to execute remote commands on the client using the **rshd** or the **NIMSH** protocol.
8. The NIM master and client must both have a minimum of 4 GB memory.
9. A reliable network, which can facilitate large amounts of NFS traffic, must exist between the NIM master and the client. The NIM master and client must be able to perform NFS mounts and read/write operations.
10. The client's hardware and software must support the AIX level that is being migrated to and meet all other conventional migration requirements.
11. The application servers, such as DB2 and LDAP, must be stopped before you run the clone rootvg command. Otherwise, the application servers do not start normally after the clone rootvg command has finished processing.

Note: If you cannot meet the alternate disk migration installation requirements 1-10, perform a conventional migration. For information on the conventional migration installation method, see “Migrating AIX” on page 405. If you cannot meet requirement 11, no migration installation is possible.

Before performing an alternate disk migration installation, you are required to agree to all software license agreements for software to be installed. You can do this by specifying the **-Y** flag as an argument to the alternate disk migration command or setting the **ADM_ACCEPT_LICENSES** environment variable to **yes**.

Alternate disk migration limitations:

These limitations apply to alternate disk migration installations.

The limitations are as follow:

- If the client's **rootvg** has Trusted Computing Base enabled, it is disabled during the migration. Trusted Computing Base is not supported on AIX 7.2.
- All NIM resources used must be local to the NIM master.

- During the migration, the client's active **rootvg** may experience a small performance decrease due to increased disk I/O, **nfsd** activity, and some CPU usage associated with **alt_disk_install** cloning.
- NFS tuning may be required to optimize performance.

Alternate disk migration installation usage:

The syntax for the alternate disk migration installation command is described.

The syntax is:

```
nimadm -l lpp_source -c NIMClient -s SPOT -d TargetDisks [ -a
       PreMigrationScript ] [ -b installp_bundle] [ -z PostMigrationScript] [
       -e exclude_files] [ -i image_data ] [ -m NFSMountOptions
       ] [ -o bosinst_data] [-P Phase] [ -j VGname ] [-Y ] [ -F ] [ -D ] [ -E
       ] [ -V ] [ { -B | -r } ]
```

Use the **nimadm** command to target the *aix1* NIM client, using the *spot1* NIM **SPOT** resource, the *lpp1* NIM **lpp_source** resource, and *hdisk1* and *hdisk2* target disks, by typing the following:

```
nimadm -c aix1 -s spot1 -l lpp1 -d "hdisk1 hdisk2" -Y
```

Use the **-Y** flag to agree to all required software license agreements for the software being installed

Clean up alternate disk migration on client:

The syntax is:

```
nimadm -C -c NIMClient -s SPOT [ -F ] [ -D ] [ -E ]
```

Wake-up volume group:

The syntax is:

```
nimadm -W -c NIMClient -s SPOT -d TargetDisks [-m NFSMountOptions ] [-z
       PostMigrationScript ] [ -F ] [ -D ] [ -E ]
```

Put-to-sleep volume group:

The syntax is:

```
nimadm -S -c NIMClient -s SPOT [ -F ] [ -D ] [ -E ]
```

Synchronize alternate disk migration software:

The syntax is:

```
nimadm -M -s SPOT -l lpp_source [ -d device ] [ -P ] [ -F ]
```

mksysb to client migration:

The syntax is:

```
nimadm -T NIMmksysb -c NIMClient -s SPOT -l lpp_source -d TargetDisks
       -j VGname -Y [ -a PreMigrationScript ] [ -b installpBundle ] [ -z
       PostMigrationScript ] [ -i ImageData ] [ -m NFSMountOptions ] [ -o
       bosinst_data ] [ -P Phase ] [ -F ] [ -D ] [ -E ] [ -V ] [ -B | -r ]
```

mksysb to mksysb migration:

The syntax is:

```
nimadm -T NIMmksysb -O mksysbfile -s SPOT -l lpp_source -j VGname -Y [
       -N NIMmksysb ] [ -a PreMigrationScript ] [ -b installp_bundle ] [ -z
       PostMigrationScript ] [ -i image_data ] [ -m NFSMountOptions ] [ -o
       bosinst_data ] [ -P Phase ] [ -F ] [ -D ] [ -E ] [ -V ]
```

Client to mksysb migration:

The syntax is:

```
nimadm -c nim_client -O mksysbfile -s SPOT -l lpp_source -j VGname -Y
      [ -N NIMmksysb ] [ -a PreMigrationScript ] [ -b installp_bundle ] [ -z
      PostMigrationScript ] [ -i image_data ] [ -m NFSMountOptions ] [ -o
      bosinst_data ] [ -P Phase ] [ -e exclude_files] [ -F ] [ -D ] [ -E ]
      [ -V ]
```

Installing alternate disk migration:

The **nimadm** command performs a migration in 12 phases.

Each phase can be executed individually using the **-P** flag. Before performing a migration in phases, you should have a good understanding of the **nimadm** process. The **nimadm** phases are as follows:

1. The master issues the **alt_disk_install** command to the client, which makes a copy of the **rootvg** to the target disks (this is Phase 1 of the **alt_disk_install** process). In this phase, **altinst_rootvg** (alternate **rootvg**) is created. If a target mksysb has been specified, the mksysb is used to create a **rootvg** using local disk caching on the NIM master.
2. The master runs remote client commands to export all of the `/alt_inst` file systems to the master. The file systems are exported as read/write with root access to the master. If a target mksysb has been specified, the cache file systems are created based on the image.data from the mksysb.
3. The master NFS mounts the file systems exported in Phase 2. If a target mskysb has been specified, the mksysb archive is restored into the cache file systems created in Phase 2.
4. If a premigration script resource has been specified, it is executed at this time.
5. System configuration files are saved. Initial migration space is calculated and appropriate file system expansions are made. The **bos** image is restored and the device database is merged (similar to a conventional migration). All of the migration merge methods are executed and some miscellaneous processing takes place.
6. All system filesets are migrated using **installp**. Any required RPM images are also installed during this phase.
7. If a **post-migration** script resource has been specified, it is executed at this time.
8. The **bosboot** command is run to create a client boot image, which is written to the client's boot logical volume (**hd5**).
9. All mounts made on the master in phase 3 are removed.
10. All client exports created in phase 2 are removed.
11. The **alt_disk_install** command is called again (phase 3 of **alt_disk_install**) to make final adjustments and put **altinst_rootvg** to sleep. The bootlist is set to the target disk (unless the **-B** flag is used). If an output mksysb has been specified, the cache is archived into a mksysb file, and made a nim mksysb resource.
12. Cleanup is executed to end the migration. The client is rebooted, if the **-r** flag is specified.

Note: The **nimadm** command supports migrating several clients at the same time.

Accessing data between the original rootvg and the new alternate disk

You can initiate data access between the original rootvg and the new alternate disk.

A volume group "wake-up" can be accomplished, on the non-booted volume group. The "wake-up" puts the volume group in a **post alt_disk_install** Phase 1 state. For example, the `/alt_inst` file system is then mounted.

The volume group that experiences the "wake-up" is renamed **altinst_rootvg**. When data access is no longer needed, the volume group can be "put to sleep."

For more information on the command flags to "wake-up" and "put to sleep", see the alt_rootvg_op man page. Once the alternate disk or rootvg is mounted, file access is the same as for any mounted file system.

Notes:

- The running operating system's version must be greater than or equal to the version of the volume group that undergoes the "wake-up." This might mean that it is necessary to boot from the **altinst_rootvg** and "wake-up" the **old_rootvg**.

  This limitation is caused by a journaled file system (JFS) log entry incompatibility. It is possible to "wake-up" a volume group that contains a more recent version, but the volume group cannot have ever been the system **rootvg**. If this was true, the volume group would have made JFS log entries that could not be interpreted by an older version **rootvg**, when the volume group was experiencing a "wake-up."

  The **alt_disk_install** command does not allow a "wake-up" to occur on a volume group with a more recent version, unless the **FORCE** environment variable is set to **yes**.
- The volume group that experiences a "wake-up" must be "put to sleep" before it can be booted and used as the **rootvg**.

  Attention: If a FORCE "wake-up" is attempted on a volume group that contains a more recent version of the running operating system, and the "waking" volume group has been a system **rootvg**, errors occur.

Running alternate disk installation by using SMIT

The procedure for running alternate disk installation using SMIT is described.

To run alternate disk **mksysb** installation, do the following:

1. At the system prompt, type the **smit alt_mksysb** fast path.
2. Type or select values in the entry fields. Press Enter after making all desired changes.

To run alternate disk **rootvg** cloning, do the following:

1. At the system prompt, type the **smit alt_clone** fast path.
2. Type or select values in the entry fields. Press Enter after making all desired changes.

Installing an alternate disk through dynamic logical partitioning

On a system that supports dynamic logical partitioning (DLPAR), you can dynamically add an adapter with disks to a running logical partition (LPAR). You can then install a new rootvg volume group to these newly added target disks using the **alt_disk_install** command with either the clone or **mksysb** option.

If you are running the **alt_disk_install** command with dynamically added target disks on an LPAR system, the following flags might be used:

-O If the target disk will be used to boot an LPAR other than the one where the operation is being executed, use the **-O** flag to reset the device information.

-B This flag prevents the **bootlist** command from being run. A general limitation of dynamically added disks is that you can not specify them as a boot device (before an initial reboot operation). If you are attempting to boot an LPAR from dynamically added disks, set the boot list in the system management services (SMS) menus.

-g This flag causes the **alt_disk_install** command to run without checking if the disk is bootable.

Dynamically added disks do not appear bootable to AIX until after a reboot operation. The user will need to verify that the newly added adapter and disks are bootable.

Examples: Installing an alternate disk

Examples of alternate disk installation are shown.

To install an alternate disk, perform one of the following procedures:

1. To clone the **rootvg** running a lower technology level to `hdisk1` and update that clone with the latest maintenance level that is on `cd0`, run the following command:

   ```
   alt_disk_copy -b update_all -l /dev/cd0 -d hdisk1
   ```

   In SMIT, use the **smit alt_clone** fast path and select **hdisk1** from the listing for Target Disk(s) to install, select the **update_all** bundle from the listings in the **Bundle to Install** field, and **/dev/cd0** from the listing in the **Directory or Device with images** field.

2. To clone the **rootvg** running 7.1.0 to `hdisk3`, then update to the latest fixes that are mounted from another system on `/710fixes`, and run a customized script named **/tmp/finish_alt_install**, run the following command:

   ```
   alt_disk_copy -b update_all -l /710fixes \
   -s /tmp/finish_alt_copy -d hdisk3
   ```

   In SMIT, use the **smit alt_clone** fast path and select **hdisk3** from the listing for Target Disk(s) to install, select the **update_all** bundle from the listings in the **Bundle to Install** field, type `/710fixes` in the **Directory or Device with images** field, and type `/tmp/finish_alt_copy` in the **Customization script** field.

3. To install an AIX **mksysb** tape that was created from a machine with the same hardware configuration as the target, to `hdisk1`, run the following command:

   ```
   alt_disk_mksysb -m /dev/rmt0 -d hdisk1
   ```

   In SMIT, use the **smit alt_mksysb** fast path and select **hdisk1** from the listing for **Target Disk(s)** to install field and select **/dev/rmt0** from the listing for **Device** or image name field.

4. To install an AIX **mksysb** image that is NFS mounted on file system **/mksysbs** to the alternate disk `hdisk2` using a customized `image.data` file and an exclude file containing **^./tmp/**, type the following command:

   ```
   alt_disk_mksysb -m /mksysbs/my_71_mksysb -i /mksysbs/my_71_image.data \
   -e /mksysbs/my_exclude_file -d hdisk2
   ```

   Using the **^./tmp/** pattern does not backup files in the `/tmp` directory, but does backup files in the `/var/tmp` directory.

   > **Note:** All files are backed up relative to the current directory. This directory is represented by a **.** (dot character). If it is important that the search match the string at the beginning of the line when excluding a file or directory, it is necessary to use a **^.** (caret followed by a dot character) as the first part of the search string, followed by the filename or directory to be excluded. The form is as follows:
   >
   > ^./*filename*
   >
   > If the file name or directory being excluded is a substring of another file name or directory, use a **^.** (caret followed by a dot character) for the search to start at the beginning of the line and the **$** (dollar symbol) to have the search finish at the end of the line.

   In SMIT, use the **smit alt_mksysb** fast path and select **hdisk2** in the **Target Disk(s) to install** field. Next, type `/mksysbs/my_71_mksysb` in the **Device** or image name field, `/mksysbs/my_71_image.data` in the **image.data** file field, and **/mksysbs/my_exclude_file** in the **Exclude** list field.

5. To "wake-up" an original rootvg, after booting from the new alternate disk, run the following command:

   ```
   alt_rootvg_op -W -d hdisk0
   ```

   The following example illustrates the output that might display when running the command discussed above:

```
# lspv
hdisk0          000040445043d9f3    old_rootvg
hdisk1          00076443210a72ea    rootvg

# alt_rootvg_op -W hdisk0

# lspv
hdisk0          000040445043d9f3    altinst_rootvg
hdisk1          00076443210a72ea    rootvg
```

At this point, the **altinst_rootvg** volume group is varied-on and the **/alt_inst** file systems are mounted.

6. To "put-to-sleep" a volume group that had experienced a "wake-up," type the following command:

   ```
   alt_rootvg_op -S
   ```

   The following example illustrates the output that might display when running the command previously discussed:

   ```
   # lspv
   hdisk0          000040445043d9f3    altinst_rootvg
   hdisk1          00076443210a72ea    rootvg

   # alt_rootvg_op -S

   # lspv
   hdisk0          000040445043d9f3    altinst_rootvg
   hdisk1          00076443210a72ea    rootvg
   ```

   The **altinst_rootvg** is no longer varied on and the `/alt_inst` file systems are no longer mounted. If necessary for the **altinst_rootvg** volume group name to be changed back to **old_rootvg**, do this task with the **-v** flag.

Using the multibos utility

The **multibos** utility allows you, as root, to create multiple instances of AIX on the same root volume group (rootvg).

The **multibos** setup operation creates a standby base operating system (BOS) that boots from a distinct Boot Logical Volume (BLV). This creates two bootable instances of BOSes on a given rootvg. You can boot from either instance of a BOS by specifying the respective BLV as an argument to the **bootlist** command, or using system firmware boot operations.

You can simultaneously maintain two bootable instances of a BOS. The instance of a BOS associated with the booted BLV is the *active* BOS. The instance of a BOS associated with the BLV that has not been booted is the *standby* BOS. Only two instances of BOS are supported per rootvg.

The **multibos** utility allows you to access, install, maintain, update, and customize the standby BOS either during setup or during any subsequent customization operations. Installing maintenance or technology level updates to the standby BOS does not change system files on the active BOS. This allows for concurrent update of the standby BOS, while the active BOS remains in production.

The **multibos** utility has the ability to copy or share logical volumes and file systems. By default, the **multibos** utility copies the BOS file systems (currently the **/**, **/usr**, **/var**, **/opt**, and **/home** directories), associated log devices, and the boot logical volume. You can make copies of additional BOS objects (see the **–L** flag). All other file systems and logical volumes are shared between instances of the BOS. Separate log device logical volumes (those not contained within the file system) are not supported for copy and will be shared.

Requirements of the multibos utility

The **multibos** utility has requirements for operating system, space, and logical volumes.

Following are the general requirements and limitations:

- The **multibos** utility is supported on AIX 5L™ Version 5.3 with the 5300-03 Recommended Maintenance package and higher versions.
- The current rootvg must have enough space for each BOS object copy. BOS object copies are placed on the same disk or disks as the original.
- The total number of copied logical volumes cannot exceed 128. The total number of copied logical volumes and shared logical volumes are subject to volume group limits.

Standby BOS setup operation

The standby BOS setup operation is described.

The **multibos** setup operation, using the **-s** flag, performs the following steps:

1. The **multibos** methods are initialized.
2. If you provide a customized `image.data` file, it is used for the logical volume attributes. Otherwise, a new one is generated. You can use the customized `image.data` file to change BOS object (logical volume or file systems) attributes. You cannot use the customized `image.data` file to add or delete BOS logical volumes or file systems.
3. The standby logical volumes are created based on **image.data** attributes. The active and standby logical volumes are marked with unique tags in the logical volume control block. The **multibos** utility uses these tags to identify copied logical volumes. If the active logical volume names are *classic* names, such as hd2, hd4, hd5, and so on, then the **bos_** prefix is prepended to create a new standby name. If the active logical volume names have the **bos_** prefix, the prefix is removed to create a new standby name.

   Note: The Logical Volume Manager (LVM) limits the maximum length of a logical volume name to 15 characters. This means that any logical volume classic name may not exceed 11 characters. You can rename logical volumes that have classic names that exceed 11 characters using the **chlv** command. If the active logical volume name already has the **bos_** prefix, then the prefix is removed in the standby name.
4. The standby file systems are created based on **image.data** attributes. The active and standby file systems are marked with unique tags in the hosting logical volume control block and `/etc/filesystems`. The **multibos** utility uses these tags to identify copied logic volumes. The **/bos_inst** prefix is prepended to the original active file system name to create the standby file system name. The standby file system name may not exceed the system's **PATH_MAX** limit. The standby file systems appear as standard entries in the active BOS `/etc/filesystems`.
5. The standby file systems are mounted.
6. A list of files that will be copied from the active BOS is generated. This list is comprised of the current files in copied active BOS file systems, less any files that you excluded with the optional exclude list (see the **-e** flag).
7. The list of files generated in the previous step is copied to the standby BOS file systems using the backup and restore utilities.
8. Any optional customization is performed. This can include installation of fileset updates or other software.
9. The standby boot image is created and written to the standby BLV using the AIX **bosboot** command. You can block this step with the **-N** flag. Only use the **-N** flag if you are an experienced administrator and have a good understanding the AIX boot process.
10. The standby BLV is set as the first boot device, and the active BLV is set as the second boot device. You can skip this step using the **-t** flag.

Automatic file system expansion

Run all **multibos** operations with the **multibos -X** flag auto-expansion feature. This flag allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks.

Booting the standby BOS

The **bootlist** command supports multiple BLVs.

As an example, to boot from disk hdisk0 and BLV bos_hd5, you would enter the following: `# bootlist –m normal hdisk0 blv=bos_hd5`. After the system is rebooted from the standby BOS, the standby BOS logical volumes are mounted over the usual BOS mount points, such as **/**, **/usr**, **/var**, and so on.

The set of BOS objects, such as the BLV, logical volumes, file systems, and so on that are currently booted are considered the active BOS, regardless of logical volume names. The previously active BOS becomes the standby BOS in the existing boot environment.

Mounting the standby BOS

It is possible to access and modify the standby BOS by mounting its file systems over the standby BOS file system mount points. The **multibos** mount operation, using the **-m** flag, mounts all standby BOS file systems in the appropriate order.

Automatic file system expansion

Run all **multibos** operations with the **multibos -X** flag auto-expansion feature. This flag allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks.

Preview option

The preview option, using the **-p** flag, applies to the setup, remove, mount, unmount, and customization operations. If you specify the preview option, then the operation provides information about the action that will be taken, but does not perform actual changes.

Unmounting the standby BOS

The **multibos unmount** operation, using the **-u** flag, unmounts all standby BOS file systems in the appropriate order.

Automatic file system expansion

The **multibos -X** flag auto-expansion feature allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks. You should execute all **multibos** operations with this flag.

Preview option

The preview option, using the **-p** flag, applies to the setup, remove, mount, unmount, and customization operations. If you specify the preview option, then the operation provides information about the action that will be taken, but does not perform actual changes.

Customizing the standby BOS

You can use the **multibos** customization operation, with the **-c** flag, to update the standby BOS.

The customization operation requires an image source (**-l** *device or directory* flag) and at least one installation option (installation by bundle, installation by fix, or **update_all**). The customization operation performs the following steps:

1. The standby BOS file systems are mounted, if not already mounted.
2. If you specify an installation bundle with the **-b** flag, the installation bundle is installed using the **geninstall** utility. The installation bundle syntax should follow **geninstall** conventions. If you specify the **-p** preview flag, **geninstall** will perform a preview operation.
3. If you specify a fix list, with the **-f** flag, the fix list is installed using the **instfix** utility. The fix list syntax should follow **instfix** conventions. If you specify the **-p** preview flag, then **instfix** will perform a preview operation.

4. If you specify the **update_all** function, with the **-a** flag, it is performed using the **install_all_updates** utility. If you specify the **-p** preview flag, then **install_all_updates** performs a preview operation.

   Note: It is possible to perform one, two, or all three of the installation options during a single customization operation.
5. The standby boot image is created and written to the standby BLV using the AIX **bosboot** command. You can block this step with the **-N** flag. You should only use the **-N** flag if you are an experienced administrator and have a good understanding the AIX boot process.
6. If standby BOS file systems were mounted in step 1, they are unmounted.

Automatic file system expansion

The **multibos -X** flag auto-expansion feature allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks. You should run all **multibos** operations with this flag.

Preview option

The preview option, using the **-p** flag, applies to the setup, remove, mount, unmount, and customization operations. If you specify the preview option, then the operation provides information about the action that will be taken, but does not perform actual changes.

Removing the standby BOS

The remove operation, using the **-R** flag, deletes all standby BOS objects, such as BLV, logical volumes, file systems, and so on.

You can use the remove operation to make room for a new standby BOS, or to clean up a failed **multibos** installation. The remove operation performs standby tag verification on each object before removing it. The remove operation will only act on BOS objects that **multibos** created, regardless of name or label. You always have the option of removing additional BOS objects using standard AIX utilities, such as **Rmlv**, **rmfs**, **rmps**, and so on. The **multibos** remove operation performs the following steps:

1. All boot references to the standby BLV are removed.
2. The bootlist is set to the active BLV. You can skip this step using the **-t** flag.
3. Any mounted standby BLVs are unmounted.
4. Standby file systems are removed.
5. Remaining standby logical volumes are removed.

Automatic file system expansion

The **multibos -X** flag auto-expansion feature allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks. You should execute all **multibos** operations with this flag.

Preview option

The preview option, using the **-p** flag, applies to the setup, remove, mount, unmount, and customization operations. If you specify the preview option, then the operation provides information about the action that will be taken, but does not perform actual changes.

Rebuilding the standby BOS boot image

The rebuild boot image operation, using the **-B** flag, enables you to rebuild the standby BOS boot image.

The new boot image will be based on standby BOS system files and written to the standby BLV. The **multibos** build boot image operation performs the following steps:

1. The standby BOS file systems are mounted, if they are not already.
2. The standby boot image is created and written to the standby BLV using the AIX **bosboot** command.

3. If the standby BOS file systems were mounted in step 1, they are unmounted.

Automatic file system expansion

The **multibos -X** flag auto-expansion feature allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks. You should execute all **multibos** operations with this flag.

Using the standby BOS shell operation

The **multibos** shell operation **-S** flag enables you to start a limited interactive chroot shell with standby BOS file systems.

This shell allows access to standby files using standard paths. For example, **/bos_inst/usr/bin/ls** maps to **/usr/bin/ls** within the shell. The active BOS files are not visible outside of the shell, unless they have been mounted over the standby file systems. Limit shell operations to changing data files, and do not make persistent changes to the kernel, process table, or other operating system structures. Only use the BOS shell if you are experienced with the chroot environment.

The **multibos** shell operation performs the following steps:

1. The standby BOS file systems are mounted, if they are not already.
2. The **chroot** utility is called to start an interactive standby BOS shell. The shell runs until an exit occurs.
3. If standby BOS file systems were mounted in step 1, they are unmounted.

Here is an example of some operations that can be performed in the **multibos** shell:

```
MULTIBOS> lppchk –v   # check system fileset consistency
MULTIBOS> installp -ug bos.games # removes bos.games
MULTIBOS> oslevel –r # reports recommended maintenance level for standby BOS
```

Automatic file system expansion

The **multibos -X** flag auto-expansion feature allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks. Start all **multibos** operations with this flag.

Additional multibos options

You can specify additional logical volumes, file systems, paging space, and so on to be copied to a standby BOS.

Preview option

The preview option, using the **-p** flag, applies to the setup, remove, mount, unmount, and customization operations. If you specify the preview option, then the operation provides information about the action that will be taken, but does not perform actual changes.

Exclude list file

You can use an optional exclude list with the setup operation. The rules for exclusion follow the pattern-matching rules of the **egrep** command.

For example, to exclude the contents of the `/tmp` directory, and avoid excluding any other directories that have **/tmp** in the path name, edit the exclude file to read as follows: `^./tmp/`

Note: All files are backed-up relative to the current working directory ("."). To exclude any file or directory for which it is important to have the search match the string at the beginning of the line, use the caret character (^) as the first character in the search string, followed by the dot character (.), followed by the filename or directory to be excluded. If the file name or directory being excluded is a substring of

another file name or directory, then use the caret character followed by the dot character (^.) to indicate that the search starts at the beginning of the line. Use the dollar sign ($) to indicate that the search stops at the end of the line.

Specifying additional BOS objects

By default, only a subset of the BOS file systems are copied to the standby BOS. These file systems currently are **/**, **/usr**, **/var**, **/opt**, and **/home**. The boot logical volume is also copied to the standby BOS. You can specify additional logical volumes, file systems, paging space, and so on to be copied to the standby BOS with the **-L** *AddFile* flag. The *AddFile* file needs to contain the logical volume names associated with the active BOS object. For example, if you intended to copy the **/mylocal** file system, then you would include the name of the logical volume that is mounted over **/mylocal** (for example, Lv01). See the **lsfs** command for details about how to match file system mount points to logical volumes. Separate log devices (those not contained within the file systems) are not supported for copy and will not be copied even if listed in the *AddFile* file.

Note: Only LVM-based objects (that is, objects associated with logical volumes) are supported for **multibos** copying.

Automatic file system expansion

The **multibos -X** flag auto-expansion feature allows for automatic file system expansion, if space is necessary to perform **multibos**-related tasks. Run all **multibos** operations with this flag.

Customizing your installation

You can customize your AIX installation. Customizing an installation requires you to edit the `bosinst.data` file and use it with your installation media.

The first time you install, the Base Operating System (BOS) installation program presents menus from which you must choose setup options. This initial installation also automatically starts a post-installation configuration program, either the graphical Configuration Assistant or the ASCII Installation Assistant.

For subsequent installations, you can change many aspects of the default BOS install program by editing the `bosinst.data` file. For example, to install the BOS without menus, you can specify that no prompts be provided. You can also customize a BOS installation to bypass Configuration Assistant or Installation Assistant and start your own configuration script. Also, the `bosinst.data` file can be used to replicate one set of installation settings on other machines. Set the CONNECTION field for obtaining the correct hard disk information for each system. For example, system administrators can create a `bosinst.data` file with settings that can be used to install all the machines they support that have the same configuration.

If you run your own configuration script from a `bosinst.data` file or from the Network Installation Management (NIM) interface, the environment that is in place at the time the script is run is a *single-user environment*. This environment is not available as a multiuser environment, and thus, there are limits to what can be run from a configuration script. The `/etc/init` file is not running, so no process management can take place. All available memory cannot be made available because the RAM file system still exists, so devices that require large amounts of memory to run might fail to configure. In addition, signal handling is not available.

Because of the single-user environment, use the following guidelines for configuration scripts:

- Base devices can be configured, but devices that require daemons or more complex configuration should be started at reboot time by adding the necessary code to the end of the `/etc/firstboot` script.
- Daemons should not be started.
- Items such as NIS configuration, which uses system resource controller (SRC) commands, should be done by creating a separate entry in the `/etc/inittab` file and running a configuration script at reboot time.

- The BOS installation process automatically creates and extends paging space based on available memory.

The `bosinst.data` file directs the actions of the BOS installation program. The file resides in the **/var/adm/ras** directory on the installed machine only, and it is not accessible on the commercial tape or the media on which you received AIX.

The `bosinst.data` file contains stanzas with variables set to default values. Each variable is on a new line, in the *Variable*=*Value* form. A blank line separates each stanza. These stanzas provide the installation program with information such as the method and type of installation, the disks in the machine, and the language used. By editing the file with an ASCII text editor, you can substitute new values for the default variables.

Another installation file, **image.data**, can also be modified and used during BOS installation. The `image.data` file contains information describing the root volume group image created during the BOS installation process. This information includes the sizes, names, maps, and mount points of logical volumes and file systems in the root volume group. The installation program also takes input from the **image.data** file regarding defaults for the machine being installed. The procedure for using the `bosinst.data` file to customize BOS installation can also be used for the **image.data** file. The modified files can be used together to override BOS installation defaults.

You can also use the instructions in this chapter to create a supplemental diskette, a CD-R, or a DVD-RAM containing a modified `preserve.list` file, which is used during a preservation Installation.

Related concepts:

"Configuring AIX" on page 85
Complete all configuration tasks that apply to your newly installed system. Two configuration tools are available to assist you.

"Installing the Base Operating System" on page 39
There are multiple ways to install the AIX base operating system.

"Installing system backups" on page 329
You can install the Base Operating System (BOS) using a system backup image, also called an *mksysb image*.

Related information:

AIX Files

"The bosinst.data file" on page 46
The content and use of the `bosinst.data` file is described.

Customizing and using the bosinst.data file

You must install the BOS before you can access and modify the default `bosinst.data` file.

You can also edit the `bosinst.data` file like any other ASCII file.

For information about the contents of the file and examples of edited files, refer to "bosinst.data file stanza descriptions" on page 46 and "Using the bosinst.data file" on page 56.

Note: If you are customizing the `/bosinst.data` file so that it becomes part of a system backup (mksysb), the **mksysb** command always updates the **target_disk_data** stanzas to reflect the current disks in the **rootvg**. If you do not want this update to occur, you must create the file `/save_bosinst.data_file`. The existence of this file is checked by the **mksysb** command, before the **target_disk_data** stanzas are updated.

To edit and use the `bosinst.data` file, use one of the following procedures:

Using a customized bosinst.data file with NIM

You can use a customized bosinst.data file for network installations.

Create one customized bosinst.data file for each client, and using Network Installation Management (NIM), define the files as NIM resources. For more information about how to use the bosinst.data file as a resource in network installations, refer to "The bosinst.data file" on page 46.

Creating and using a supplementary bosinst.data media

Use this procedure to create the supplementary media and use it in future installation.

1. Customize the bosinst.data file and create a signature file by completing the following steps:
   a. Use the **mkdir** command to create a directory called /tmp/mycd: mkdir /tmp/mycd.
   b. Use the **cd** command to change your directory to the /tmp/mycd directory: cd /tmp/mycd.
   c. Copy the /var/adm/ras/bosinst.data file to /tmp/mycd.
   d. Copy the /var/adm/ras/bosinst.data file to /tmp/mycd.
   e. Edit the bosinst.data file with an ASCII editor to customize it.
   f. Create a signature file: echo data > signature.
   g. Change the permissions on the file using the following command: # chmod 777 *.
2. Create the customized media by completing the following steps:
   a. Use the **cd** command to change your directory to the / directory.
   b. Create the customized media using the following command (where /dev/cd1 varies depending on your CD or DVD writer device): # mkcd -d /dev/cd1 -r /tmp/mycd
3. Use the customized media for installation by completing the following steps:
   - If you have only one media drive and you are installing from CD or DVD, complete the following:

   a. Insert the installation media in the media drive of the machine where you are installing AIX.
   b. Boot the machine from the installation media.
   c. Type 311 at the BOS welcome screen. You will be prompted to insert the customized media.
   d. Insert the customized media. The BOS installation program uses the bosinst.data file on the media, rather than the bosinst.data file on the boot media. For more information on the bosinst.data file, see "The bosinst.data file" on page 46.

   - If you are performing a network installation or tape **mksysb** installation, or if you have more than one media drive, complete the following:

   a. Insert the customized media in the media drive of the machine where you are installing AIX.
   b. Boot the machine from the network or a tape.

      Note: You can boot from a CD or DVD and use a tape for the installation. However, during a tape boot, you cannot use the CD and DVD drives to supply customized information.
   c. Type 311 at the BOS welcome screen. The installation continues for a non-prompted installation, or the menus display for a prompted installation.

      The BOS installation program uses the bosinst.data file on the media, rather than the bosinst.data file from the boot media. For more information on the bosinst.data file, refer to "The bosinst.data file" on page 46.

Creating and using a supplementary bosinst.data diskette

You can create a supplementarybosinst.data diskette to use for customized installations.

Complete the following process to create the supplementary diskette:

1. Customize the bosinst.data file and create a signature file by completing the following steps:
   a. Use the **mkdir** command to create a directory called /tmp/mydiskette: mkdir /tmp/mydiskette
   b. Use the **cd** command to change your directory to the /tmp/mydiskette directory: cd /tmp/mydiskette

c. Copy the /var/adm/ras/bosinst.data file to /tmp/mydiskette.
d. Edit the bosinst.data file with an ASCII editor to customize it.
e. Create a signature file: echo data > signature

2. Create the diskette and use it for installation by completing the following steps
   a. Back up the edited bosinst.data file and the new signature file to diskette with the following command: ls ./bosinst.data ./signature | backup -iqv.

      OR

      If you create a bundle file named mybundle, back up the edited bosinst.data file, the new signature file, and the bundle file to diskette with the following command: ls ./bosinst.data ./signature ./mybundle | backup -iqv
   b. Insert the diskette in the diskette drive of the target machine you are installing.
   c. Boot the target machine from the installation media (DVD-ROM, or network) and install the operating system. The BOS installation program uses the diskette file, rather than the default bosinst.data file shipped with the installation media. For more information on the bosinst.data file, see "The bosinst.data file" on page 46.

Installing AIX on a system with multiple disks

Save time on AIX installations by specifying the disks on which you want the system installed.

In general, if you do not specify the disk (root volume group) on which you want the AIX system installed, the operating system is installed on a disk that was previously installed with AIX. If you have many disks that contain data volume groups, and these data volume groups are discovered before the previous root volume group is found, the installation can be delayed until a suitable disk is found. First specify the disk on which you want to install the system, and you will save time. You can specify the installation disk by using one of the following methods:

- Specify the installation disk in the bosinst.data file by physical location code (PHYSICAL_LOCATION) or physical volume identifier (PVID):
  1. To determine the physical location on a running system, type:

     ```
     lsdev -F "name physloc" -l hdisk
     ```
  2. To determine the physical volume identifier on a running system, type:

     ```
     lsattr -E -O -a pvid -l hdisk
     ```
  3. If you are using a fibre-channel disk for the installation, you can use the following command in the **bosinst.data** file:

     ```
     SAN_DISKID=worldwide_portname//lun_id
     ```
- Specify the installation disk in the [2] file from DVD, or through a network installation.
  1. For a network installation, specify the installation disk in the bosinst.data file by typing the following command:

     ```
     nim -o bos_inst -a bosinst_data=value ...
     ```
  2. For an installation from DVD, specify the installation disk in the bosinst.data file by following the procedures at "Customizing and using the bosinst.data file" on page 82.

If you do not specify the characteristics of the disk in the bosinst.data file on the target system, the installation disk is chosen based on the flags in the control_flow stanza of the bosinst.data file. Unless you specify EXISTING_SYSTEM_OVERWRITE=no, the first suitable root volume group is chosen for the installation. For overwrite or preserve installations, any root volume group is acceptable. For migration, the volume group must be installed with a version of the operating system that can be migrated to the level being installed. If you specify EXISTING_SYSTEM_OVERWRITE=no in the control_flow stanza of the bosinst.data file, then the installation goes to the first unused disk.

2. bosinst.data

Installing content for the man command

The documentation for AIX commands, files and libraries in **man** command format for English are contained on both the *AIX operating system* DVD and on the *AIX Documentation* DVD. Translated versions of this documentation are contained only on the *AIX Documentation* the DVD.

Note: These filesets are not translated into all languages.

The fileset names for the AIX commands, files and libraries in **man** command format are titled as follows:

- **infocenter.man.XX_XX.commands**
- **infocenter.man.XX_XX.files**
- **infocenter.man.XX_XX.libs**

Where XX_XX is the language indicator for that fileset (for example, EN_US). While performing installs from either the *AIX operating system* DVD or the *AIX Documentation* DVD, select filesets such as those in the list above for the languages in which you wish to view **man** command documentation.

Configuring AIX

Complete all configuration tasks that apply to your newly installed system. Two configuration tools are available to assist you.

Depending on which type of console you are using, one of the following usually begins automatically after installation:

- Configuration Assistant for graphics consoles
- Installation Assistant for ASCII consoles

Notes:

- If your system was installed by a network installation server, the Configuration Assistant or Installation Assistant does not display when the BOS installation program completes.

  If your system was installed using a system backup image, or if your BOS installation was customized, or if you selected migration installation from AIX, the Configuration Assistant or Installation Assistant might not display when the BOS installation program completes.
- The Configuration Assistant and the Installation Assistant do not contain the tasks needed to configure your machine as a server. If you need to configure your system for a specific resource, refer to the documentation pertaining to that resource.
- If your terminal type is not set, the first menu displayed by the ASCII Installation Assistant requires you to enter your terminal type (TTY). If you enter a terminal type that is not valid, this menu redisplays until a valid type is entered.

  If you enter a valid terminal type that does not match your terminal, the next screen displayed might be unreadable. In this case, press the break key sequence to return to the Set Terminal Type screen. For most terminal types, the break key sequence is `Ctrl-C`.

Related concepts:

"Customizing your installation" on page 81
You can customize your AIX installation. Customizing an installation requires you to edit the `bosinst.data` file and use it with your installation media.

Configuring AIX with the Configuration Assistant

On a system with a graphical interface, the newly installed BOS reboots and the Configuration Assistant guides you through the configuration tasks.

If there are outstanding software license agreements that must be accepted before you can continue to use the machine, the Configuration Assistant prompts you to view and accept these agreements.

The Configuration Assistant guides you through the following configuration tasks:

- Set or verify system date and time.
- Set password for administrator (root user).
- Configure network communications (TCP/IP).

  Note: To configure your machine as an NFS server, refer to Configuring an NFS server in *Networks and communication management*.
- Manage Software.
- Exit the Configuration Assistant.

The Manage Software option allows you to perform software management tasks immediately after a BOS installation. The following options are available:

- List installed software
- Install additional software
- List software licenses with license text

If you select **List installed software**, the following options are available:

- List automatically installed Software – Displays a list of all installed packages
- List optionally installed software – Displays a list of all optional software that was selected to be installed during BOS installation

If you select **Install additional software**, the following options are available:

- Install by bundles – Allows you to select from a list of software bundles to install additional software, such as the Mozilla Software Bundle or a User-Defined Software Bundle
- Selective install – Allows you to select a specific package or set of packages to install

The graphical interface for the Configuration Assistant provides step-by-step instructions for completing each configuration task. The tasks are presented to you in a logical sequence. Complete all configuration tasks before you use your system.

When you exit the Configuration Assistant, the guide asks you whether you want to start Configuration Assistant again the next time you restart the operating system. After exiting the Configuration Assistant, users can begin logging in to and using AIX.

To access the Configuration Assistant later, type `configassist` on the command line.

Configuring AIX with the Installation Assistant

On a system with an ASCII interface, the newly installed BOS reboots, and the Installation Assistant guides you through the configuration tasks.

You must have root user authority to use the Installation Assistant. To access the Installation Assistant later, type `install_assist` on the command line. You can also access it from a graphics system through the SMIT **smit assist** fast path.

If there are outstanding software license agreements that must be accepted before you can continue to use the machine, the Installation Assistant prompts you to view and accept these agreements.

The Installation Assistant guides you through the following configuration tasks:

- Set the system date and time for your time zone.
- Set a root user account password to restrict access to system resources.
- Configure network communications.

- Install software applications.
- Using SMIT (information only).
- Tasks Completed - Exit to Login.

The Install software applications option allows you to perform software management tasks immediately after a BOS installation. The following options are available:

- Install and Update Software
- Add License Passwords for Applications
- Show Installed License Agreements

If you select **Install and Update Software**, the following menu displays:

```
                        Install and Update Software

Move cursor to desired item and press Enter.

  Install Software
  Update Installed Software to Latest Level (Update All)
  Install Software Bundle
  Update Software by Fix (APAR)
  Install and Update from ALL Available Software
```

You can also access this SMIT menu by using the **install_update** fast path.

Related information

The following are links to information related to Configuring AIX.

If you are installing from DVD-ROM, or would like more information about installing optional software, refer to "Preparing to install optional software products and service updates" on page 338.

Troubleshooting your installation

Find tactics for isolating installation and configuration problems, and their solutions.

Troubleshooting an installation from a system backup

Troubleshoot common problems when installing from a system image created with the **mksysb** command.

Installing when booting a system backup fails

If a backup tape fails to boot, you can still install by using a **mksysb** image stored on the tape.

Boot the machine from the product media (Volume 1 if there is more than one volume), then install the backup from Maintenance mode. For instructions on booting, refer to "Installing the Base Operating System" on page 39. Follow the instructions to the point when the Welcome to the Base Operating System Installation and Maintenance screen displays.

Booting system backup from the product media:

Follow this procedure to boot a system backup from the product media.

Complete the following steps when the Welcome screen is displayed:

1. Choose the **Start Maintenance Mode for System Recovery** option.
2. Choose the **Install from a System Backup** option.
3. Choose the drive containing the backup tape.

The system reads the tape and begins the installation.

4. Do not remove the disk from the media drive.

   The system installs the kernel and device support required on the target system from the disk.

5. Return to step 9 on page 334 in the Installing a System Backup on the Source Machine procedure and continue the instructions for installing the backup.

   Note: The **Use Maps** option is not supported in Maintenance Mode. For more information on the maps options in Maintenance Mode, refer to "Installing a system backup on the source machine" on page 331.

Configuring mksysb image on system backup tapes

Use the **mksysb** command to ensure that the boot image, BOS Installation/Maintenance image, and the table of contents image are created with a tape **block_size** value of 512.

Bootable **mksysb** tapes comprise the following images:

- Boot image
- BOS Installation/Maintenance image
- Table of contents image
- System backup image

The system backup image is the actual backup of the files in the rootvg in all JFS-mounted file systems.

The boot image, BOS Installation/Maintenance image, and the table of contents image must be created with a tape **block_size** value of 512. The **mksysb** command ensures that the block size is 512 when these images are created. There are no restrictions on the block size used for the fourth (system backup image) on the tape. The block size of the system, before it was temporarily set to 512, is used for the fourth image on the tape.

The value of the block size must be saved in the **/tapeblksz** file in the second image on the tape. The second and fourth images are stored in backup/restore format. Again, **mksysb** ensures the correctness of the tapes created by using the **mksysb** command.

If there are problems with the **bosinst.data** file, the **image.data** file, or the **tapeblksz** file, these files can be restored from the second image on the tape and checked. These files, as well as commands necessary for execution in the RAM file system (when running in maintenance mode after booting from the tape), are stored in the second image.

Restoring a file from the second image or tape:

Follow these steps to restore a file from the second image.

1. Be sure the tape block size is 512 by entering the following command: **# lsattr -E -l rmt0**.

   If the block size is not correct, use the following command to set it to 512: **# chdev -l rmt0 -a block_size=512**.

2. Make sure the tape is rewound. If the tape is not rewound, enter the following command: **# tctl -f /dev/rmt0 rewind**
3. Extract the necessary files by entering: `# restore -xvq -s2 -f /dev/rmt0.1`*filename*

   Note: The filename should be the full path, and always preceded with a `.` (dot character), such as `./tapeblksz`.

4. Rewind the tape by entering: `# tctl -f /dev/rmt0 rewind`
5. Change the block size back to its original value, if necessary.

Troubleshooting problems with installation from mksysb backup

These troubleshooting tips apply to reported problems with installations from a **mksysb** image.

- Check that you have sufficient free blocks in the file systems to write temporary files.
- Check that each file system has at least 500 blocks free when the **mksysb** backup image is made. The system needs workspace in each file system when installing from a **mksysb** backup image.

  Note: Depending on the type of data or files in the file system, you might need additional blocks free. For example, if the file system has a lot of small files, an extra 4 KB is automatically allocated to allow for metadata expansion.
- Check that you are using the correct tape type for the density setting that you selected.
- Check that the tape is *not* write-protected.
- Clean the tape drive at the recommended intervals and use only approved data-grade tapes (not video tapes for 8 mm).
- Check that 7206 4-mm Digital Audio Tape (DAT) tape drives are using only DAT tapes marked with the Dataphone Digital Services (DDS) symbol. Any other DAT tapes (for example, voice grade) cannot be used.
- Check the **/smit.log** file for any errors from SMIT.
- Check that your **mksysb** backup image contains an `image.data` file. If you create the **mksysb** backup image through SMIT, it is done automatically. If you run **mksysb** from the command line, you must either run the **mkszfile** command first, or use the **-i** flag with the **mksysb** command.

Troubleshooting migration installation

The following offers solutions for problems that can occur during a migration installation.

Troubleshooting boot logical volume errors

References for responding to errors indicating that the boot logical volume is not large enough is described.

If you receive errors indicating the boot logical volume is not large enough, see "Interpreting installation-related system and error messages" on page 95.

Troubleshooting insufficient disk space for migration

At the beginning of a migration installation, the system verifies that there will be enough space to attempt the migration. If there is not enough disk space, a message explains how much is needed.

You must now reboot the machine from the media containing your current version of AIX, and make more space available in the **rootvg** volume group. After you do this, attempt the migration again.

You can use the following options for adding additional disk space for the migration installation:

- Add another disk to the **rootvg** volume group, using either the SMIT **smit extendvg** fast path or the **extendvg** command.
- Move any user-data logical volumes from the **rootvg** volume group to another volume group. You can use either the SMIT **smit cplv** fast path or the **cplv** command to move individual logical volumes to another volume group's disk. It is a good idea to have only system logical volumes in the **rootvg**, and have user-data logical volumes in other volume groups.

  After you use the **cplv** command, you must remove the original logical volumes with the **rmlv** command. If the moved logical volume contains a file system, you must modify its corresponding entries in the **/etc/filesystems** file to reflect the new logical volume name.

  For more detailed information about manipulating logical volumes and volume groups, refer to Logical Volumes in *Operating system and device management*.
- Remove unneeded logical volumes (and file systems) from the **rootvg**. Run the **lsvg -l rootvg** command to see all the logical volumes in the **rootvg** volume group. The only logical volumes that

must be in the **rootvg** are: hd2, hd3, hd4, hd5, hd6, hd8, and hd9var. The hd1 (**/home**) logical volume can be located in another volume group if necessary.

The hd7 (system dump) logical volume is not needed because the paging space logical volume (hd6) is used. The migration code automatically removes this logical volume if space is needed, but you can remove it ahead of time with the following commands:

```
sysdumpdev -P -p /dev/hd6
rmlv -f hd7
```

- If you cannot find extra space in your **rootvg**, you might have to do a *preservation* installation instead of a migration installation to AIX. A preservation installation saves all the "non-system" logical volumes and file systems (for example, **/home**), but removes and re-creates the following logical volumes: hd2, hd3, hd4, hd5 and hd9var.

  If you do a preservation installation, you must reinstall any applications that were installed in your **rootvg** after the preservation installation has completed. You must also reconfigure devices, as well as re-create users and groups. For more information about a preservation installation, see “Installing the Base Operating System” on page 39.

After you have released enough space, reboot from your installation media, and try the migration installation again. You must have at least 8 MB of free disk space to complete the migration installation.

If there is insufficient space to complete the migration installation during the BOS installation process, a message similar to the following is displayed at the end of the installation:

```
An error occurred while migrating packages.

Some packages have not been installed.

Please see /var/adm/ras/devinst.log for details or perform an overwrite or
   preservation install.
```

If space limitations prevent the migration of all software that is usually automatically migrated, the installation program attempts to install the software that is usually installed for a Preservation or Overwrite installation. If there is still not enough disk space available, the minimum set of software required to support the use of the system is installed.

If there is not enough space to migrate all of the usually migrated software, a collection of software called a Migration Bundle will be available when you install additional software later. If the minimum set of software is installed, or if the installation is not performed from a graphics console, a Graphics_Startup Bundle is created. Before installing either of these bundles, create additional disk space on the machine you want to install. For more information about installing software bundles and migrating or installing optional software products, refer to “Optional products and service updates” on page 335. “Maintaining optional software products and service updates” on page 343 describes how to remove software from the system to release disk space.

Troubleshooting alternate disk installation errors

The following are error messages you might encounter during alternate disk installation.

If you receive either of the following error messages, see “Interpreting installation-related system and error messages” on page 95.

- `0505-113 alt_disk_install: No target disk name provided.`
- `0505-117 alt_disk_install: Error restoring image.data file from mksysb image.`

Troubleshooting other problems with alternate disk installation

You might encounter one of these problems with alternate disk installation.

Symptom: You have run the **alt_disk_install** command or used the SMIT menus to either clone or install a **mksysb** image on an alternate disk. However, you now want to remove the definition so you can use the disk to run the **alt_disk_install** command again or use the disk for another purpose.

Action: *Do not run* the **exportvg** command. The **exportvg** examines the logical volumes on the disk (now called by their rootvg names: **hd1**, **hd2**, **hd3**, and so on) and tries to remove their corresponding entries from the `/etc/filesystems` file. This action removes the real file system stanzas from your running system and causes boot problems if you reboot with the missing stanzas.

Use the **alt_disk_install -X** command to remove the **altinst_rootvg** name from the database. This removes only the ODM information from the CuDv database, so the **lspv** command shows the disk(s) as no longer belonging to **altinst_rootvg**. It also resets your bootlist to the boot disk on which the hd5 boot logical volume resides. You can still boot from the **altinst_rootvg**, because the volume group, logical volume, and file system information remain on the disk. However, you must set your bootlist to the **altinst_rootvg** boot disk.

Troubleshooting after a BOS installation

The following are troubleshooting tips for issues that might arise following a BOS installation.

The Configuration Assistant or Installation Assistant will not display when the BOS installation program completes if your system was installed by a network installation server.

Configuration Assistant and Installation Assistant do not contain the tasks needed to configure your machine as a server. If you need to configure your system for a specific resource, refer to the documentation pertaining to that resource.

If your terminal type is not set, the first menu displayed by the ASCII Installation Assistant requires you to enter your terminal type (tty). If you enter a terminal type that is not valid, this menu redisplays until a valid type is entered.

If you enter a valid terminal type that does not match your terminal, the next screen displayed may be unreadable. In this case, press the break key sequence to return to the Set Terminal Type screen. For most terminal types, the break key sequence is Ctrl-C.

Troubleshooting a system that does not boot from the hard disk

Follow this procedure to access a system that will not boot from the hard disk.

If a **mksysb** backup tape fails to boot, read "Troubleshooting an installation from a system backup" on page 87 for instructions.

This procedure enables you to get a system prompt so that you can attempt to recover data from the system or perform corrective action that will enable the system to boot from the hard disk.

Notes:

1. This procedure is intended only for experienced administrators who have knowledge of how to boot or recover data from a system that is unable to boot from the hard disk. Most administrators should not attempt this procedure but instead should follow local problem-reporting procedures.
2. This procedure is not intended for administrators who have just completed a New Installation, because the system will not contain data that needs to be recovered. If you are unable to boot from the hard disk after completing a New Installation, follow your local problem-reporting procedures.

The following steps summarize the procedure for accessing a system that will not boot.

1. Boot the system from Volume 1 of the BOS media or a bootable tape.
2. Select **Maintenance Options**.

3. Recover data or perform corrective action using the system prompt.

Preparing to access a system that does not boot

You must meet these prerequisites before attempting to access a system that will not boot from the hard disk.

Confirm the following:

- Your system cannot be booted from the hard disk.
- All hardware is installed.
- AIX Base Operating System (BOS) is installed.
- Your system unit is set to Off.

Accessing the system if unable to boot from the hard disk

Use this procedure if you are unable to boot from the hard disk.

The beginning of this procedure is similar to the one you used to install the base operating system. You will, however, use the maintenance screens instead of the installation screens to complete this procedure.

1. Turn on all attached external devices, such as terminals, media drives, tape drives, monitors, and external disk drives *before* turning on the system unit. Do not turn on the system unit until step 5. Turning on the external devices first is necessary so that the system unit can identify them during the startup (boot) process.
   - If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.
   - If you are not booting from a network device, go to step 3.
2. Insert Volume 1 of the installation media into the tape or media drive. Some media drives have a removable disc caddy, while others have a sliding drawer. If the media drive on your system has a sliding drawer, place the media in the drawer and push the drawer in. If the media drive on your system does not have a sliding drawer, insert the media into the disc caddy and then insert the caddy into the CD-ROM drive.

   Notes:

   a. You may find that on specific hardware, the tape drive door will not open while the system unit is turned off. If you have trouble opening the tape drive door during installation, use the following procedure:
      1) Turn the system unit on.
      2) Insert the BOS tape (insert Volume 1 if you received more than one volume).
      3) Turn the system unit off and wait 30 seconds.
   b. On some models that have a door to the tape drive, there may be a waiting period of up to three minutes before the tape drive door opens after you have pressed the button to open the tape drive. Some models also require that the button for the tape drive door be held in the pressed position for a few seconds before the tape drive door will open.
   c. On some models, the eject button must be pressed for at least 2 seconds to eject media that is already in the disc caddy.
3. If you are not using an ASCII terminal, skip to step 5. If you are using an ASCII terminal, set the communications options as follows:
   - Line Speed (baud rate) = 9600
   - Word Length (bits per character) = 8
   - Parity = no (none)
   - Number of Stop Bits = 1
   - Interface = RS-232C (or RS-422A)
   - Line Control = IPRTS

Set the keyboard and display options as follows:

- Screen = Normal
- Row and Column = 24x80
- Scroll = jump
- Auto LF (line feed) = off
- Line Wrap = on
- Forcing Insert = line (or both)
- Tab = field
- Operating Mode = echo
- Turnaround Character = CR
- Enter = return
- Return = new line
- New Line = CR
- Send = page
- Insert Character = space

Note: If your terminal is an IBM 3151, 3161, or 3164, press the Ctrl+Setup keys to display the Setup Menu and follow the on screen instructions to set these options. If you are using some other ASCII terminal, refer to the appropriate documentation for information about how to set these options. Some terminals have different option names and settings than those listed here.

4. Turn the system unit power switch to the On position. The system begins booting from the installation media. If your system is booting from tape, it is normal for the tape to move back and forth. After several minutes, `c31` is displayed in the LED.

   If you have more than one console, each terminal and directly attached display device (or console) might display a screen that directs you to press a key to identify your system console. A different key is specified for each terminal displaying this screen. If this screen is displayed, then press the specified key on the device to be used as the system console. The system console is the keyboard and display device used for installation and system administration. Press a key on only one console.

5. Type `3` to select **Start Maintenance Mode for System Recovery** from the Welcome to the Base Operating System Installation and Maintenance screen when it displays.

   Note: If you customized the `bosinst.data` file in your installation media to specify a nonprompted installation, the installation and maintenance screens are not displayed. The system instead reboots from the installation media using the settings already defined in the `bosinst.data` file. To access the installation and maintenance screens, override the nonprompted mode. You can do this when three zeros are displayed on the screen. When you observe the three zeros, type `000` (zeros) and press Enter at the terminal.

   You can select **88** to display help on this or any subsequent screen.

   After you have selected the **Start Maintenance Mode for System Recovery** option, the Maintenance screen displays.

6. Select option `1`, **Access a Root Volume Group**, from the Maintenance screen. The Warning screen displays.
7. Read the information displayed on the Warning screen. When you are ready to continue, type `0` and press Enter. The Access a Root Volume Group screen displays.
8. Select the option for the root volume group whose logical volume information you want to display. The Access a Root Volume Group screen lists all of the volume groups (root and otherwise) on your system. After entering your selection, the Volume Group Information screen displays.

Note: Reviewing the disk and location code information on the Volume Group Information screen enables you to determine whether the volume group you selected was the root volume group. You can return to the Access a Root Volume Group screen if the choice you made was not the root volume group. If you have not chosen a root volume group, you cannot continue beyond the Volume Group Information screen.

9. Select one of the options from the Volume Group Information screen and press Enter. Each option does the following:

Item	Description
Choice 1	**Access this volume group and start a shell.** Selecting this choice imports and activates the volume group and mounts the file systems for this root volume group before providing you with a shell and a system prompt.
Choice 2	**Access this volume group and start a shell before mounting file systems.** Selecting this choice imports and activates the volume group and provides you with a shell and system prompt before mounting the file systems for this root volume group.
Choice 99	Typing 99 returns you to the Access a Root Volume Group screen.

After you select either choice**1** or **2**, a shell and system prompt display.

10. Take appropriate measures to recover data or take action (such as using the **bosboot** command) to enable the system to boot normally.

Troubleshooting a full /usr file system

Use this procedure for troubleshooting a full /usr file system.

To release space in a full /usr file system, complete one or more of the following tasks:

- Type `installp -c all` to commit all updates and release space in the /usr file system.
- If the system is not a Network Installation Management (NIM) system serving a Shared Product Object Tree (SPOT), enter `/usr/lib/instl/inurid -r` to remove client information for root file system installations. For information about NIM and SPOTs, see "Using the SPOT resource" on page 253 in the NIM Resources section.

  Note: You must not run the **inurid** command to free space if you have shared /usr workload partitions or plan to have shared /usr workload partitions.
- Remove software that you do not need. See "Maintaining optional software products and service updates" on page 343.

Viewing BOS installation logs

Information saved in BOS installation log files might help you determine the cause of installation problems.

To view BOS installation log files, type `cd /var/adm/ras` and view the files in this directory. One example is the **devinst.log**, which is a text file that can be viewed with any text editor or paged.

Viewing BOS installation logs using SMIT

You can use the SMIT fast path to view some logs in the /var/adm/ras directory.

To view some logs in the /var/adm/ras directory, you can use the following SMIT fast path:

```
smit alog_show
```

The resulting list contains all logs that are viewable with the **alog** command. Select from the list by pressing the F4 key.

Viewing BOS installation logs with the alog command

You can use the **alog** command to view some logs in the /var/adm/ras directory.

To view some logs in the /var/adm/ras directory, type:

```
alog -o -f bosinstlog
```

Interpreting installation-related system and error messages

These messages might appear during the installation of AIX.

Information about most messages is provided in the following format:

Item	Description
System Message	The system message is displayed in **bold** type.
Explanation	Describes what is likely to have caused the system message to be displayed.
System Action	Describes what the system does after the message is displayed.
User Action	Suggests a possible resolution to the problem suggested by the system message.

Note:

Multiple messages can have the same explanation, system action, and user action.

0516-404 allocp: Not enough resources available to fulfill allocation. Either not enough free partitions or not enough physical volumes to keep strictness. Try again with different allocation characteristics.

0516-788: extendlv: Unable to extend logical volume

0503-008 installp: There is not enough free disk space in file system **/usr** (506935 more 512 byte blocks are required.) An attempt to extend this file system was unsuccessful. Make more space available, then retry this operation.

Item	Description
Explanation	There is not enough space to complete the installation.
System Action	The installation cannot begin until the problem is resolved.
User Action	You have several options: • Select fewer filesets than the number originally selected for installation. OR • Extend the root volume group to another disk. Type: `extendvg rootvg hdisk` *`Number`*, where *Number* is the number of the specified disk. OR • Remove user-defined file systems to release space in the **rootvg** file system. OR • Follow the instructions in "Troubleshooting a full `/usr` file system" on page 94.

BOS Install: After saving all the data from the previous system into **/tmp**, it was discovered that there will not be enough free space in **/tmp** to make the boot image. Please reboot in normal mode and increase the size of **/tmp** or reduce the number of files to save as listed in the **/etc/preserve.list** file.

Item	Description
Explanation	During a preservation installation, files listed in the**/etc/preserve.list** file were copied to the **/tmp** file. After doing so, there was not enough room in **/tmp** to create the boot image.
System Action	Installation cannot continue.
User Action	Reboot in normal mode and increase the size of **/tmp** or reduce the number of files to be saved.

BOS Install: You chose to create logical volumes mapped exactly as they were on the previous disks, but there are no map files specified in the **image.data** file.

Item	Description
Explanation	On system backup restore, **EXACT_FIT = yes** was specified in the **image.data** file, but no map files were specified in the **image.data** file.
System Action	Nonprompted mode is terminated. The user is prompted.
User Action	Run the **mkszfile** command with the **-m** option before creating the system backup tape. OR Do not specify **EXACT_FIT = yes** in the `image.data` file.

The boot logical volume (`hd5`) must be at least 24 MB. The system you are installing has a boot logical volume smaller than this, and the system does not have enough free contiguous physical partitions on *diskname* to increase the size of the boot logical volume. Please reboot in normal mode and correct this problem, or restart the installation and choose an overwrite install. Use the lspv -M *diskname* command to see the current allocation map of the disk.

OR

Error: No space available to create a larger boot logical volume. In order to proceed with this installation the size of the boot logical volume (`hd5`) must be increased to 24 MB. At this time there are not *N* contiguous physical partitions available on the boot disk (*diskname*) for recreating the larger boot logical volume. You must free up this space by removing or relocating one or more logical volumes or file systems from *diskname*. Use lspv -M *diskname* to see its current partition allocation map.

Item	Description
Explanation	The boot logical volume (**blv**), logical volume `hd5`, must be greater than 24 megabytes. If your system had disks less than 4 gigabytes in size in the root volume group, your boot logical volume may only be 4 megabytes. You might experience this failure during preservation or migration installations. Overwrite installations create the boot logical volume with a minimum size of 24 megabytes. If free partitions contiguous to `hd5` are available or if another location on the disk contains `hd5` is identified, the installation process increases the size of `hd5` and continues. Only the disk that currently contains the boot logical volume is checked for additional partitions in order to increase the size of the boot logical volume. Other disks in the rootvg are not checked.
System Action	You will be prompted to reboot in normal mode from the existing rootvg and increase the boot logical volume, or restart the installation and choose an overwrite install.

Item	Description

User Action	Only a system administrator with root authority should attempt to increase the boot logical volume. To increase the boot logical volume, follow the process described below:
	
	If you received this error, then your partition size is less than 8 megabytes, and you must increase the number of partitions in `hd5` (boot logical volume). You can check your partition size as follows:
	1. Type the following:
	`# lsvg rootvg`
	2. Look for the field: `PP SIZE:`
	3. Obtain the current number of partitions in `hd5`, as follows:
	`# lslv hd5`
	4. Look for the field: `LPs:`
	5. Your boot logical volume must contain enough partitions such that:
	• `PP SIZE` multiplied by `LPs` is greater than or equal to 24.
	• The partitions for the boot logical volume must be contiguous.
	
	If there were free partitions available next to `hd5` or at some other location on the disk that contains `hd5`, the installation process would have increased the size of `hd5`, and continued.
	
	To view the current allocation map (free and used partitions) of a disk, use the command:
	`# lspv -M` *`diskname`*

Item	Description

User Action, continued	If there are not enough contiguous free partitions, you must increase the size of the boot logical volume (hd5) using one of the options described below, and rerun the installation. The options for increasing the boot logical volume size are as follows:
	• If a user-created logical volume or file system follows hd5 on the disk (check the allocation map), and has free partitions, you can back up, remove, re-create, and restore the logical volume.
	• If there is another disk in the rootvg, that has enough contiguous free partitions, then you could move hd5 to the other disk with the following steps:
	1. Verify that the disk you plan to move hd5 to is bootable by using the command:
	`bootinfo -B diskname`
	– If 1 is returned, the disk is bootable.
	– If 0 is returned, the disk is not bootable.
	2. Find the free contiguous partitions you need on the other disk by viewing the allocation map with the command:
	`lspv -M diskname`
	3. Create a map file to use when re-creating hd5. For example, if you want to re-create hd5 on hdisk2, on partitions 88 and 89, use the command:
	`echo "hdisk2:88-89" > your_MAP_file`
	4. Remove the existing hd5:
	`rmlv -f hd5`
	5. Create the new hd5:
	`mklv -y hd5 -t boot -m your_MAP_file rootvg 2`
	The 2 represents the number of partitions and can vary as needed.
	Note: If the **mklv** command moves hd5 to a new location, you must run the following command:
	`echo ":C:C:C" | /usr/lpp/bosinst/blvset -d /dev/hdiskN`
	Where *C* is the message, locale, and keyboard (respectively) and *hdiskN* is the disk that contains hd5.
	6. Run the **mkboot** command to clear the boot record from the disk that previously contained hd5 (boot logical volume). For example, if hd5 was previously on hdisk0, use the command:
	`mkboot -d /dev/hdisk0 -c`
	7. Use the **bosboot** command to re-create the boot image and boot record on the new disk. For example, if hd5 was re-created on hdisk2, use the command:
	`bosboot -a -d /dev/hdisk2`

Item	Description

User Action, continued	1. Change the bootlist of your system to boot from the new disk. To see the current bootlist, use the command:
	`bootlist -m normal -o`
	OR
	If your previous `hd5` was on `hdisk0`, the output might be:
	`hdisk0`
	To change the bootlist to use `hdisk2`, use the command:
	`bootlist -m normal hdisk2`
	If there were additional items in your bootlist, add them after `hdisk2`, with spaces separating each item.
	2. If there were no errors, reboot your system.
	3. If you encountered this error when installing a **mksysb** on a system other than the system it was created on (cloning), then you might be able to use a customized `image.data` file to increase the size of `hd5`.
	The **vg_data** stanza contains the size of the physical partitions in the **PPSIZE** field. Use this information to determine how many partitions are needed for `hd5`. The **lv_data** stanza for `hd5` contains the fields for the number of logical partitions (**LPs**), the number of physical partitions (**PP**), and the minimum number of logical partitions required for the logical volume (**LV_MIN_LPS**). These fields must be set to the number of partitions needed.
	See "Creating and using a supplementary bosinst.data diskette" on page 83 for information on putting an `image.data` file on diskette and a `bosinst.data` file.
	If the source machine had no free partitions, and the target machine has the same disk size, then you might need to install using the shrink option, as well as the customized `image.data` file.

BOS Install: Could not create boot image.

Item	Description
Explanation	The **bosboot** command failed.
System Action	The boot image was not created.
User Action	Check the `/var/adm/ras/bosinstlog` file for errors (**alog -o -f bosinstlog | pg**). This log is updated by appending, so make sure you check the last entry.

The bosinst.data file does not specify any bootable disks.

Item	Description
Explanation	The `bosinst.data` file does not specify any bootable disks.
System Action	Non-prompted mode is terminated. The user is prompted.
User Action	When the system prompts, select bootable disks to install on. OR Add a bootable disk to the `bosinst.data` file target_disk_data stanzas.

The bosinst.data file specified doing a migration install, but there is no existing root volume group.

Item	Description
Explanation	A BOS installation method of **migration** was specified in the `bosinst.data` file, but the existing volume group is at a lower level.
System Action	This error only occurs during a nonprompted BOS installation. The installation menus are displayed.
User Action	Respond to the menu prompts to complete the installation.

The bosinst.data file specified doing either a migration or a preservation install, but there is no existing root volume group.

Item	Description
Explanation	A BOS installation method of **migrate** or **preserve** was specified in the **bosinst.data** file, but no root volume group was found.
System Action	This error only occurs during a non-prompted BOS installation. The installation menus are displayed.
User Action	Respond to the menu prompts to complete the installation.

The data file did not specify enough disk space to contain the operating system.

Item	Description
Explanation	Non-prompted mode was specified, and there were not enough disks specified in the `bosinst.data` file to hold the operating system.
System Action	Non-prompted mode is terminated. The user is prompted.
User Action	When the system prompts, select disks to install on. OR Add more **target_disk_data** stanzas to `bosinst.data` file.

Duplicate **lv_data** stanzas specified in the **image.data** file. The installation cannot continue because data may be lost.

Item	Description
Explanation	An **lv_data** stanza was duplicated in the `image.data` file.
System Action	Installation cannot continue.
User Action	Correct the problem and try the installation again.

Duplicate **fs_data** stanzas specified in the **image.data** file. The installation cannot continue because data may be lost.

Item	Description
Explanation	An **fs_data** stanza was duplicated in the `image.data` file.
System Action	Installation cannot continue.
User Action	Correct the problem and try the installation again.

The following disks failed the preliminary diagnostic tests: <disk name>

bosset: No hard disks can be accessed.

Item	Description
Explanation	The listed disks failed pretest.
System Action	The system initiated a diagnostic pretest on the specified disk.
User Action	Run full diagnostics on the specified disks.

Disks specified in bosinst.data do not define a root volume group.

Item	Description
Explanation	Non-prompted mode was specified. The install method was set to **preserve** or **migrate**, and the disks specified in **bosinst.data** do not define a root volume group.
System Action	Non-prompted mode is terminated. The user is prompted.
User Action	When the system prompts, select a root volume group to install on. OR Specify disks in the `bosinst.data` file that define a root volume group.

Encountered an unrecoverable error.

Item	Description
Explanation	The menus subsystem encountered an unrecoverable error.
System Action	The menu is restarted.
User Action	None

The **image.data** file contains no **vg_data** stanza for rootvg. The installation cannot continue.

Item	Description
Explanation	The **image.data** file is incomplete.
System Action	Installation cannot continue.
User Action	Use the default **image.data** file supplied with product media.

image.data has invalid logical volume data. Cannot continue.

Item	Description
Explanation	The system could not parse the logical volume data stanzas in the `image.data` file.
System Action	Installation cannot continue.
User Action	Use the default `image.data` file supplied with product media.

image.data has invalid file system data. Cannot continue.

Item	Description
Explanation	The system detected invalid file system data stanzas in the `image.data` file.
System Action	Installation cannot continue.
User Action	Use the default `image.data` file supplied with product media.

0516-366 putlvodm: Volume group rootvg is locked. Try again.

0516-788: extendlv: Unable to extend logical volume.

Item	Description
Explanation	You interrupted the installation of your optional software.
System Action	When an installation is interrupted, the system sometimes locks the root volume group.
User Action	Unlock the root volume group. Then attempt the installation procedure again. To unlock a root volume group: 1. Log in with root authority. 2. Type `chvg -u rootvg` 3. Type `smit_install` and attempt to install your optional software products again.

installp: An error occurred during bosboot processing.

Correct the problem and rerun.

0301-52 bosboot: not enough file space to create: **/tmp/disk.image**.

OR

0301-152 bosboot: not enough file space to create: **/tmp/unix**.

Item	Description
Explanation	The **bosboot** command was unable to finish processing because of insufficient space in **/tmp**.
System Action	The **bosboot** process is interrupted. The error message, the amount of disk space required, and the available disk space are displayed. The disk space displayed indicates the number of 1024 KB blocks required.
User Action	Release space in the **/tmp** file system or extend the **/tmp** file system. Continue or restart the installation process. To resize the **/tmp** file system and complete the installation, do the following: 1. Note the error message preceding this one. Either the message `bosboot verification starting` or `bosboot process starting` will precede this message. 2. Change directories to **/tmp**. List the files and determine which files can be deleted. If there is sufficient space available, go to step 6. If you need to expand the **/tmp** file system, continue with this procedure. 3. Type `smit chfs` 4. Select the `/tmp` file system from the displayed list. 5. Add the additional block space required. The **smit chfs** command requires disk space to be defined in 512 KB blocks. Double the required disk space displayed in the system message. 6. If the message `installp: An error occurred during bosboot processing` was displayed after the message `bosboot verification starting`, rerun the installation procedure. OR If the message `installp: An error occurred during bosboot processing` was displayed after the message `bosboot process starting`, enter `installp -C`. 7. Continue the installation process.

installp: An error occurred during bosboot processing.

Correct the problem and rerun.

301-155 bosboot: Invalid or no boot device specified.

Item	Description
Explanation	A device specified with the **bosboot -d** command is not valid. The **bosboot** command was unable to finish processing because it could not locate the required boot device. The **installp** command calls the **bosboot** command with **/dev/ipldevice**. If this error does occur, it is probably because **/dev/ipldevice** does not exist. **/dev/ipldevice** is a link to the boot disk.
System Action	The bosboot process is interrupted.
User Action	Determine if the link to the boot device is missing or incorrect, correct the error and complete the installation process. To identify the boot device and complete the installation: 1. To identify the boot disk, enter `lslv -m hd5`. The boot disk name displays. 2. Create a link between the boot device indicated and the `/dev/ipldevice` file. Enter: `ln /dev/`*`boot_device_name`*` /dev/ipldevice` (An example of *boot_device_name* is **rhdisk0**.) 3. If the message `installp: An error occurred during bosboot processing` was displayed after the message `bosboot verification starting`, rerun the installation procedure. OR If the message `installp: An error occurred during bosboot processing` was displayed after the message `bosboot process starting`, enter `installp -C`. Continue the installation process.

Missing image.data file. The tape does not contain a valid install image.

Item	Description
Explanation	The system could not find an `image.data` file.
System Action	Installation cannot continue.
User Action	The most likely cause of this error is the tape is bad. Try a different tape.

0512-0016 mksysb: Attempt to create a bootable tape failed: **bosboot -d /dev/device -a** failed with return code xxx.

OR

0512-0016 mksysb: Attempt to create a bootable tape failed: **mkinsttape /dev/device** failed with return code xxx.

Item	Description
Explanation	The *xxx* return code indicates the error: **5 OR 1** Not enough space in one or more of three file systems: • **/** must have at least 500 1KB blocks. • **/tmp** must have at least 7400 1KB blocks. • **/usr** must have at least 4000 1KB blocks. **11** Defective tape. **42 OR 45** Either the **/usr/lib/boot/unix** file is corrupted (may be 0 length) or the link to **/unix** is missing. **48** Cannot write to the tape drive or cannot read **/dev/blv**. This is probably caused by an incorrect density setting for the tape drive. It could also be caused by either a hardware problem with the tape drive or by dirty heads on the drive.
System Action	The **mksysb** command failed to make a bootable tape.

Item	Description
User Action	The return code *xxx* indicates the action required: **5 OR 1** Check the **/**, **/tmp**, and **/usr** file systems and create more space as required. **11** Replace the defective tape. **42 OR 45** Either restore the **/usr/lib/boot/unix** file from the original tape or create the missing link. **48** Check the tape drive settings and clean the heads.

There are no disks available on this system.

Item	Description
Explanation	No hard disks are configured on the system. Consequently, the only functioning menu option is the maintenance option.
System Action	Installation cannot begin until the problem is resolved.

User Action	You have several options:
	• Select **Maintenance** (option 3) from the Welcome to Base Operating System Install Menu, and select the **Limited Function Maintenance Shell**. Verify that no disks were configured by entering the following command:
	`lsdev -Cc disk`
	To determine if there were configuration errors, enter the command:
	`cfgmgr -v 2>1 | tee /tmp/cfgmgr.out`
	You can use the **cat** command to view the `/tmp/cfgmgr.out` file, and look specifically for errors in configuration of disks. The file can be copied to diskette media using either the **dd** or **pax** commands, and moved to a running system for ease of viewing.
	OR
	• Turn off the system and check the following on systems with SCSI devices:
	– Check all SCSI devices to ensure that all SCSI addresses are unique.
	– Make sure the SCSI cards are properly terminated.
	– If external SCSI devices are in use, make sure that the SCSI chain is terminated and that the devices are turned on.
	– Check the SCSI cabling and connections.
	– Reboot and attempt the installation again.
	OR
	• Turn off the system and check the following on systems with IDE devices:
	– Check all IDE devices to ensure that all IDE master and slave settings are unique per controller. If only one IDE device is connected to a controller, it must be set to master. If an ATA device (disk) and an ATAPI device (CD-ROM or tape) are connected to the same controller, the ATA device must be set to the master device and the ATAPI device must be set as the slave device.
	– Check the IDE cabling and connections.
	– Reboot and attempt the installation again.
	OR
	• Boot from the diagnostics and check the hard disks.
	OR
	• Follow your local problem-reporting procedures.

There are no disks on this system which can be booted.

Item	Description
Explanation	The system could not find any bootable disks on the system.
System Action	Installation cannot continue.
User Action	Some third-party disks are not bootable. If a disk should be bootable but is not, run diagnostics.

You chose to install only onto disks in the existing root volume group and those not in any volume group. There are not enough of those disks to contain the **mksysb** image.

Item	Description
Explanation	The **EXISTING_SYSTEM_OVERWRITE** field in **bosinst.data** was set to **yes**, and prompt was set to **no**, and there were not enough disks on the system that contained the root volume group or contained no volume group.
System Action	Non-prompted mode is terminated. The user is prompted.
User Action	Use **target_disk_data** stanzas to specify the disks to install on, set **SHRINK** to **yes** in the **image.data** file, or at the BOS Install prompt set the **EXISTING_SYSTEM_OVERWRITE** in the **bosinst.data** file to **any**. This allows any disks to be used for the installation. **Attention:** If **EXISTING_SYSTEM_OVERWRITE** is set to **any**, user volume groups might be overwritten. OR When the system prompts, select disks on which to install or select to shrink the file systems.

You chose to install only onto disks which are not contained in a volume group, but there are not enough of those disks to contain the mksysb image.

Item	Description
Explanation	The **EXISTING_SYSTEM_OVERWRITE** field in **bosinst.data** was set to **no**, and prompt was set to **no**, and there were not enough disks on the system that contained a volume group.
System Action	Non-prompted mode is terminated. The user is prompted.
User Action	If you want the system to select the disk to install on, use the **target_disk_data** stanzas to specify the target disks and set the appropriate setting for **EXISTING_SYSTEM_OVERWRITE**, leave **EXISTING_SYSTEM_OVERWRITE** blank in the **bosinst.data** file, or set SHRINK to yes in the **image.data** file and retry the installation. OR When the system prompts, select disks on which to install.

0505-113 alt_disk_install: No target disk name provided.

Item	Description
Explanation	This message is displayed in the following situations: • You did not enter a target disk. • The disk that was specified as the target disk has a volume group already associated with it. Running the **lspv** command should show the word `None` by disks that do not have a volume group associated with them, which is what the **alt_disk_install** command checks. • The target disk (or disks) specified are not bootable. The **alt_disk_install** command runs **bootinfo -B** *disk_name* on each disk specified in the target disk list. If any one **bootinfo -B** command returns a 0, then the disk is not bootable, and it cannot be used as a target disk for the **alt_disk_install** operation.

0505-117 alt_disk_install: Error restoring **image.data** file from **mksysb** image.

Item	Description
Explanation	This message is displayed when you are trying to install a **mksysb** image from tape. The **alt_disk_install** command first checks the second image on the tape for a `./tapeblksz` file, which contains the block size in which the **mksysb** image was created. The **mksysb** command creates this file and puts it in the second image on the tape. The first three images of a **mksysb** tape are always created at a 512 byte block size. The **mksysb** image (the fourth image on the tape) can be created at another block size. If the **alt_disk_install** command cannot restore the `./tapeblksz` file from the second image, the block size will remain what it was when the **alt_disk_install** command was started. It will attempt to restore the `./image.data` file from the **mksysb** image. If this block size does not match the block size in which the **mksysb** image was created, the restore fails, and the **alt_disk_install** command produces this error.

The size of a disk is too large for the running kernel.

Item	Description
Explanation	This message is displayed when the BOS menus load to indicate that the size of one of the selected disks for installation is larger than 1 TB (1048576 MB).
System Action	Non-prompted mode is terminated. The user is prompted.
User Action	When prompted, select a smaller disk. OR Restart the installation with AIX media (product or **mksysb** image) that supports booting the 64 bit kernel.

Could not determine kernel type.

Item	Description
Explanation	This message is displayed when the installation program cannot determine the kernel type at the time of the installation.
System Action	The installation will be paused for troubleshooting.
User Action	Contact your service representative for troubleshooting.

Could not determine the largest disk size.

Item	Description
Explanation	This message is displayed when the installation program cannot determine the largest disk size on the system.
System Action	The installation will be paused for troubleshooting.
User Action	Verify that the data in the **target_disk_data** stanzas in the `bosinst.data` file are correct and follow the guidelines for stanza validation as specified in the `bosinst.template.README` file.

The size of the Logical Volume (logical volume name) is larger than the size supported by the running kernel.

Item	Description
Explanation	This message is displayed when the installation program detects that one of the Logical Volumes being created is larger than 1 TB (1048576 MB) and the running kernel is not the 64 bit kernel.
System Action	The installation will be paused for troubleshooting.
User Action	Restart the installation with AIX media (product or **mksysb** image) that supports booting the 64 bit kernel to prevent data loss or errors.

The size of one of the Logical Volumes in the rootvg disk or disks is larger than the size supported by the running kernel.

Item	Description
Explanation	This message is displayed when the installation program detects that one of the Logical Volumes on the root volume group about to be imported is larger than 1 TB (1048576 MB) and the running kernel is not the 64 bit kernel.
System Action	The installation will be paused for troubleshooting.
User Action	Restart the installation with AIX media (product or **mksysb** image) that supports booting the 64 bit kernel to prevent data loss or errors.

Network Installation Management

AIX Network Installation Management (NIM) allows you to manage the installation of the Base Operating System (BOS) and optional software on one or more machines.

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 includes 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*. Any machine that receives NIM resources is a client, although the same machine can also be a server in the overall network environment.

Most installation tasks in the NIM environment are performed from one server, called the *master*. 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.

NIM concepts

To use all the available features in NIM, you should understand various components of AIX installation.

NIM objects

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.

NIM machines

The types of machines that can be managed in the NIM environment 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.

Operating NIM on client machines:

There are unique operations to initialize the different client configurations. NIM checks 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.

Table 11. Machine Configuration

NIM Operation	Standalone	Diskless	Dataless	WPAR
bos_inst	x			
dkls_init		x		
dtls_init			x	
diag	x	x	x	
cust	x			x
fix_query	x			x
lppchk	x			x
maint	x			x
maint_boot	x			
reset	x	x	x	x
check	x	x	x	x
showlog	x	x	x	x
reboot	x	x	x	x
activate				x
chwpar				x
create				x
deactivate				x
destroy				x
lswpar	x			x
syncwpar	x			x

Defining NIM clients:

You can use the NIM **define** operation to define stand-alone, diskless, and dataless clients.

The client system can be either *managed* or *unmanaged*. A managed client is associated with a managing system that controls the client.

Managed clients use the network-boot and power-control capabilities of the **dsm.core** fileset when the file is installed. For example, You can request a maintenance boot of the client without accessing the managing system to request a network-boot.

When the **dsm.core** fileset is installed the additional capabilities of the managed clients in comparison with the unmanaged clients, is as follows:

- Performs a network-boot and boot in maintenance mode by using the following command:
  ```
  nim -o maint_boot -a boot_client=yes
  ```
- Performs a network-boot and installs the client by using the following command:
  ```
  nim -o bos_inst -a boot_client=yes
  ```
- Boots or reboots the client with the **nim -o reboot** parameter.
- Opens a virtual console **xterm**, when using the **-a open_console** parameter on selected **NIM** operations.
- Defines and uses virtual optical devices to allow the VIOS clients to mount an ISO image from a virtual CD.

Managed clients require an **mgmt_profile** attribute. This attribute specifies the management object that controls the client. It also provides the client identifier (**lpar_id** or **blade slot**).

To set the **mgmt_profile** attribute, use the **mgmt_source** and **identity** attributes. The **mgmt_profile** can be set directly, but it must not be merged with the **mgmt_source** and **identity** attributes of the define operation.

Depending on the controlling system, the **mgmt_source** and **identity** attributes provide the following information:

- If the client is managed by Hardware Management Console (HMC), the **mgmt_source** attribute must be a CEC or a VIOS object and the identity must be the **lpar** identifier of the client.
- If the client is managed by an Integrated Virtual Machine (IVM), the **mgmt_source** attribute must be an IVM object and the identity must be the **lpar** identifier of the client.
- If the client is managed by a Blade Center Management Module (BCMM), the **mgmt_source** attribute must be a BCMM object and the identity must be the **blade slot** of the client.

For managed clients, include the real network-adapter-hardware address in the **if** attribute to activate the network-boot capabilities.

The definition of the CEC,HMC IVM, VIOS, and BCMM management object is described in

To define a stand-alone, diskless, or dataless client, enter the command-line syntax as follows:

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

where the following attributes are required:

Item	Description
-t *MachineType*	Specifies the type of machine being defined. Valid values are **stand-alone**, **diskless**, **dataless**, and **wpar**.

Item	Description
-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 IP address of the client 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 must be used to define the network. *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 a network-boot of the client. The actual adapter hardware address must be used to enable the use of network-boot capabilities provided to managed systems by **dsm.core**. *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 netboot_kernel=*Value*
Specifies the kernel type of the client. Valid values are **up** for uniprocessor machines, **mp** for multiprocessor machines, and **64** for 64-bit processors. The default value is **64**.

-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 required values and optional values:

Value 1 = NetworkType **(required)**
Specifies the values **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 can 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 is 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 must 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.

-a net_settings=*Value1 Value2*
Specifies the speed and duplex settings to use for the client's ethernet adapter during a network installation, and requires a sequence number when specified. When initiating an install and reboot of a client, NIM will set these parameters in the bootlist.

Value1= auto, 10, 100, or 1000
The default is value is *100*.

Value2 = auto, half, or full
The default value is *full*.

For example:

```
nim -o change -a net_settings1="10 half" jellyfish
```

-a connect=*Value*
Specifies the communicating service used by the NIM client for remote execution of NIM commands. Value options are **shell** (for RSH) and **nimsh**. The default setting is **connect=shell**.

-a mgmt_profile=*Value1 Value2*
Stores managing system information for a NIM client that is managed by another NIM client, and requires a sequence number when specified. A mgmt_profile setting is required for operations on WPAR clients which must be performed by a managing system. Two values are required for this attribute.

Value1 Specifies the name of the NIM client which manages this client.

Value2 Specifies the name of the system as known on the managing system. This may be different than the name of the NIM object for the system.

This attribute must not be used in conjunction with the **mgmt_source** attribute and should be used for WPAR clients.

-a mgmt_source=*Value*
Stores managing system information for a NIM client that is managed by a hardware control point. The value must point to an existing NIM object, which can be one of the following objects:

- A `VIOS` object for logical partitions (LPARs) that are attached to a Virtual I/O Server object.
- A `CEC` object for LPARs that are defined on a server.
- An `IVM` object for LPARs that are attached to an Integrated Virtualization Manager (IVM).
- A `BCMM` object for blades.
- A `NAS_FILER` object that can be used for system management of an LPAR.
- An `HMC` object that can be used for systems management of an LPAR.
- A `PowerVC` object that can be used for system management of an LPAR.

-a identity=*Value*
Stores the client identifier information for a NIM client that is managed by a hardware control point. The value must be the client identifier on the hardware control point, which is one of the following:

- The LPAR identifier for LPARs
- The blade slot on the BCMM for blades

-a dump_port=*Value*
Specifies the TCP/IP port number that is used to transfer dump images from the diskless (`<= Missing "=" value`), and dataless clients to the dump resource server. This port number is used by a dump resource server, and otherwise has no meaning. The default value is **32600**.

-a vlan_tag=*Value*
Specifies the virtual logical area network identifier used for VLAN tagging. The ID identifies which VLAN the Ethernet frame belongs. With this ID, the network administrator can organize the client's communication logically rather than to the subnet. This value is used by NIM to preform a network boot on a client by using the specified VLAN tag. Configuration of the VLAN tag communication must be handled using this value before using NIM. Valid value is from 0 to 4094. The vlan_tag and vlan_pri together makes up the VLAN tag Ethernet frame header.

-a vlan_pri=*Value*
Specifies the virtual logical area network priority used for VLAN tagging. The priority identifies which VLAN the Ethernet frame belongs. With this priority, the network administrator can organize the client's communication logically rather than to the subnet. This value is used by NIM to preform a network boot on a client by using the specified VLAN tag. Configuration of the VLAN tag communication must be handled using this value before using NIM. The valid value is from 0 to 4094. The vlan_tag and vlan_pri together makes up the VLAN tag Ethernet frame header.

Standalone NIM clients:

Standalone NIM clients can be booted and operated 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.

Booting a standalone client from the network:

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**). If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.

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. For more information about these and other operations, see "Using NIM operations" on page 257.

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.

Defining a machine as diskless or dataless has the following advantages:

- **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 BOS 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.

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:

Item	Description
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. 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.

Item	Description
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 dataless clients. It is required that either a **root** resource or a **shared_root** resource be allocated for diskless clients.
dump	Defined as a parent directory in which client dump directories are maintained. When a dump resource is allocated to a client, NIM creates a subdirectory identified by the client's name for the client's exclusive use. After initialization, the client uses this subdirectory to store any dump images it creates. Note that such dumps are firmware-assisted. A **dump** resource is optional 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.
shared_root	Defined as a / (**root**) directory shared by one or more diskless clients. All clients that use a **shared_root** resource will mount the same directory as the / (root) file system. Because STNFS is used to mount the **shared_root**, any change made by a client to its root filesystem is kept local and is invisible to other clients or to the server of the **shared_root** resource. Any change a client makes to its root filesystem is also lost when the client is rebooted. It is required that either a **shared_root** resource or a **root** resource be allocated for diskless clients. A **shared_root** resource cannot be allocated by 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.

Initializing 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.

Item	Description
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.
	• If a **root** resource is used instead of a **shared_root** resource, then 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 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 dump is configured once the client is running if a dump resource is allocated to the client 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 uninstall software on a diskless or 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. Because 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. For more information about the **cust** and **maint** operations, see “Using NIM operations” on page 257.

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 unless there are **shared_root** objects that were defined from the **SPOT** and are allocated for client use.

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** and to any **shared_root** directory that has been defined from the **SPOT**.

When NIM is used to uninstall 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 also automatically "removes" the files from the client systems.
2. The **root** files of the software are removed from the client **root** directories and from any **shared_root** directory that has been defined from the **SPOT**.

NIM also provides a **sync_roots** operation to perform consistency verification and correction to ensure that the client **root** directories, and the **shared_root** directories defined from the **SPOT**, match the **root** parts stored in the **SPOT**.

Defining WPAR clients:

Workload partitions are software-based partitions that provide separate regions of application space within a single instance of the operating system.

System WPARs are a unique instance of AIX with associated file systems and security domains. The operations to manage the WPAR are performed by a managing system that shares its operating system kernel with the WPARs on that system. Application WPARs are isolated process environments that do not have separate operating system environments (file systems and security domains). Only system WPARs may be managed by NIM. For more information on workload partitions, see IBM Workload Partitions for AIX.

Workload partitions (WPAR) are represented in NIM as the wpar machine class.A WPAR can either be managed or unmanaged. A managed WPAR is associated with the managing system that hosts the WPAR. The managing system can perform management tasks to create, start, stop, and remove the WPAR. A WPAR must have a sequenced mgmr_profile attribute. This attribute identifies the name of the NIM object for the managing system and the local WPAR name on the managing system. For example, if the `goslin` WPAR is created on the `ranger` system, the mgmt_profile1 attribute would be `ranger goslin`. Operations on the `goslin` WPAR (that must be run through the managing system) are executed on the `ranger` system.

A NIM administrator can use several NIM commands to perform WPAR-system-management tasks. For more information about managing WPAR clients, see “Managing WPAR clients.”

The following optional resources are managed by NIM to support WPAR clients:

resolv_conf
Contains the name-server IP addresses and a network domain name.

Managing WPAR clients:

WPAR clients are installed and started differently than stand-alone clients.

The WPAR file systems are created on a managing system. The WPAR is started through the processes which are running on the managing system and sharing the kernel of the managing system. The file systems for the WPAR can be remote.

Creating WPAR clients:

A WPAR has unique file system, device, network, security, and resource characteristics. A WPAR can be created with either default or customized characteristics. These characteristics can be changed after the WPAR is created. You can use NIM resources and flags to customize the client when it is created.

After a WPAR system is defined with a mgmt_profile, use the `create` operation to create the WPAR on the managing system:

```
nim -o create -a attr=value... WPARName
```

The following optional resources can be used to create WPAR clients:

wpar_spec
A WPAR specification file. For a comprehensive description of the format and permitted contents of a specification file, see the `/usr/samples/wpars/sample.spec` file.

devexports
An alternate file that can be used as the master device-exports file. This file must match the format of a device-exports file (**devexports**). If a devexports resource is not allocated when the WPAR is created, the `/etc/wpars/devexports` file on the managing system is used.

secattrs
The initial security-attributes file used when the WPAR is created. If a secattrs resource is not allocated at the time the WPAR is created, the `/etc/wpars/secattrs` file on the managing system is used. The secattrs resources are not used when creating a WPAR from a savewpar backup image.

savewpar
A WPAR backup image. If a WPAR backup image is allocated, this image is used to create the WPAR through the **restwpar** command on the managing system. The savewpar image must be at the same operating system level as the managing system.

resolv_conf
A resolv_conf resource represents a file that contains valid `/etc/resolv.conf` file entries. These entries define the Domain Name Protocol name-server information for local resolver routines. To use the `/etc/resolv.conf` file from the managing system, include the -r flag with the cmd_flags attribute value.

fb_script
A script which runs when the WPAR is booted for the first time.

image_data
An image_data resource is a file which contains detailed logical-volume and file-system characteristics used to create a system. An image_data resource should only be used by advanced system adminstrators who need to control these settings.

The following optional attributes can be used to create WPAR clients:

cmd__flags
Flags that pass directly to the command that is used to create the WPAR on the managing system. If a savewpar resource is allocated to the WPAR, the **restwpar** command is run. If savewpar resource is not allocated, the **mkwpar** command is run.

Starting WPAR clients:

Use the **activate** operation to start a managed WPAR client on the managing system.

For more information about the **activate** operation, see “Using the NIM activate operation” on page 257.

Stopping WPAR clients:

Use the **deactivate** operation to stop a managed WPAR client on the managing system.

For more information about the **deactivate** operation, see "Using the NIM deactivate operation" on page 269.

Checking WPAR clients:

Use the **check** operation to check the status of a WPAR client.

For more information about the **check** operation, see "Using the NIM check operation" on page 267.

Listing characteristics of WPAR clients:

Use the **lswpar** operation to check the characteristics of a managed WPAR client on the managing system or the characteristics of WPARs on a named standalone client.

For more information about the **lswpar** operation, see "Using the NIM lswpar operation" on page 274.

Change characteristics of WPAR clients:

Use the **chwpar** operation to change the characteristics of a managed WPAR client on the managing system or the characteristics of WPARs on a named stand-alone client.

For more information about the **chwpar** operation, see "Using the NIM chwpar operation" on page 268.

Synchronizing software on WPAR clients:

Use the **syncwpar** operation to synchronize the software levels of a managed WPAR with its managing system or the managed WPARs on a named stand-alone client.

For more information about the **syncwpar** operation, see "Using the NIM syncwpar operation" on page 279.

Creating backups of WPAR clients:

Use the **define -t savewpar** operation to create a backup image of a managed WPAR client.

For more information about the **define -t savewpar** operation, see "Using the NIM define operation" on page 270.

Installing software on WPAR clients:

When a WPAR shares the `/usr` and `/opt` file systems with a managing system, the recommended WPAR default file-system configuration permits only the following software installation operations: synchronization operations and relocatable installation packages that can be installed outside of the `/usr` and `/opt` file systems.

If a WPAR is configured to have detached and writable `/usr` file systems, use the **nim -o cust** operation to install software on a WPAR client or stand-alone client.

For more information about the **nim -o cust** operation, see "Using the NIM cust operation" on page 268.

NIM Commands

The following references provide more information on NIM commands.

The nim_master_setup command:

The **nim_master_setup** command installs the **bos.sysmgt.nim.master** fileset, configures the NIM master, and creates the required resources for installation, including a **mksysb** system backup.

The **nim_master_setup** command uses the **rootvg** volume group and creates an `/export/nim` file system, by default. You can change these defaults using the **volume_group** and **file_system** options. The **nim_master_setup** command also allows you to optionally not create a system backup, if you plan to use a **mksysb** image from another system. The **nim_master_setup** usage is as follows:

```
Usage nim_master_setup: Setup and configure NIM master.
       nim_master_setup [-a mk_resource={yes|no}]
 [-a file_system=fs_name]
 [-a volume_group=vg_name]
 [-a disk=disk_name]
 [-a device=device]
 [-B] [-v]

-B    Do not create mksysb resource.
-v    Enable debug output.

Default values:
mk_resource = yes
file_system = /export/nim
volume_group = rootvg
device = /dev/cd0
```

The nim_clients_setup command:

The **nim_clients_setup** command is used to define your NIM clients, allocate the installation resources, and initiate a NIM BOS installation on the clients.

The **nim_clients_setup** command uses the definitions in the **basic_res_grp** resource to allocate the necessary NIM resources to perform a **mksysb** restore operation on the selected clients. The usage for **nim_clients_setup** is as follows:

```
Usage nim_clients_setup: Setup and Initialize BOS install for NIM clients.
       nim_clients_setup [-m mksysb_resource]
 [-c] [-r] [-v] client_objects
-m    specify mksysb resource object name -OR- absolute file path.
-c    define client objects from client.defs file.
-r    reboot client objects for BOS install.
-v    Enables debug output.
```

Note: If no client object names are given, all clients in the NIM environment are enabled for BOS installation; unless clients are defined using the **-c** option.

Other NIM commands reference:

The *Commands Reference* set provides reference information about the NIM commands, AIX operating system commands, and commands for other licensed programs for end users, system administrators, and programmers.

These books contain examples and descriptions of the commands and their available flags. The command entries are arranged in alphabetic order:

- *Commands Reference, Volume 1* contains commands ac through cx
- *Commands Reference, Volume 2* contains commands da through hy
- *Commands Reference, Volume 3* contains commands ib through mw
- *Commands Reference, Volume 4* contains commands na through rw
- *Commands Reference, Volume 5* contains commands sa through uu

- *Commands Reference, Volume 6* contains commands va through yp

For example, *Commands Reference, Volume 3* contains reference information for the NIM **lsnim** command. The *Commands Reference, Volume 4* contains reference information for the following NIM commands:

- **nim**
- **nimclient**
- **nimconfig**
- **nimdef**
- **niminit**
- **nimquery**
- **nim_update_all**

Resolving host names with NIM

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 *Networks and communication management*.

Naming 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, hyphen (`-`), exclamation mark (`!`), and the underscore character (_).
- Invalid NIM name characters include the dot character, all shell metacharacters, all file system metacharacters, and all regular expression metacharacters.

NIM environment control

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.

The **control** attribute indicates four control states. 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 control states are as follows:

Item	Description
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.

Using NIM with Dynamic Host Configuration Protocol (DHCP)

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.

Use host names whenever possible when defining NIM machine objects.

Configuring NIM

You can use several methods for performing basic NIM operations and configuration tasks.

You can perform basic NIM operations and configuration tasks using the following methods:

- System Management Interface Tool (SMIT)
- Command line

Note: 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**.

Configuring the NIM master and creating basic installation resources

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 with SMIT, or the command line.

Note:

1. Using an AIX Version 5 or Version 6 or Version 7 **lpp_source** to install filesets on an AIX Version 4 client through NIM, is not supported. If installing Version 5 or Version 6 or Version 7 filesets on a Version 4 system is necessary, the user can NFS export the **lpp_source**, mount it on the client, and then use the **installp** command or **geninstall** command to perform the installation procedures.
2. 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 1 GB of available disk space. If such space is not available, see "Using client machines as resource servers" on page 163, and "Defining an lpp_source on DVD-ROM versus hard disk" on page 151.

Configuring the NIM master and creating basic installation resources using SMIT:

Use this procedure for configuring the NIM master and creating basic installation resources using SMIT.

1. Insert the *AIX Volume 1* media into the appropriate drive of the designated master machine.
2. To install the `bos.sysmgt.nim.master` fileset, enter the **smit install_latest** fast path.
3. Using the LIST option, select `/dev/cd0` for the INPUT device/directory for software.
4. Specify **bos.sysmgt.nim.master** 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` 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:

1. Depending on the speed of your machine, creating the basic NIM resources could be a lengthy process.
2. 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.
3. 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 in the previous procedure). Continuing from the last point of failure results in faster configuration for experienced administrators.

Configuring the NIM master and creating basic installation resources from the command line:

Use this procedure for configuring the NIM master and creating basic installation resources from the command line.

1. Insert the *AIX Volume 1* media into the appropriate drive of the designated master machine.
2. To install the `bos.sysmgt.nim.master` fileset from the disk, enter: `# installp -agXd /dev/cd0 bos.sysmgt.nim.master`
3. To configure the NIM master with the following configuration, enter: `# nimconfig -a netname=network1 -a pif_name=tr0 -a ring_speed1=16 -a platform=chrp -a netboot_kernel=mp`

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

   Note: For additional attribute information, see the **nimconfig** command.

4. 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 jfs2 -g rootvg -a size=$((2000*400)) \
   -m /export/lpp_source -A yes -p rw -t no
   ```

5. To mount the file system, enter: `# mount /export/lpp_source`
6. 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
   ```

7. 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 jfs2 -g rootvg -a size=$((2000*200)) \
   -m /export/spot -A yes -p rw -t no
   ```

8. To mount the file system, enter: `# mount /export/spot`
9. The **SPOT** resource will be installed from images in the image source (in this case, the **lpp_source** that was created in step 6). 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
   ```

10. 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 jfs2 -g rootvg -a size=$((2000*150)) \
    -m /export/dd_resource -A yes -p rw -t no
    ```

11. To mount the file system, enter: `# mount /export/dd_resource`
12. 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**, which is required for diskless and dataless clients unless a **shared_root** resource (for diskless clients only) is used instead, enter:

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

    - To create the **shared_root** resource named **shared_root1**, which is required for diskless clients unless a root resource is used instead, enter:

      ```
      # nim -o define -t shared_root -a server=master \
      -a location=/export/dd_resource/shared_root1 \
      -a spot=spot1 shared_root1
      ```

    - To create the **dump** resource named **dump1** (optional), 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
```

Using NIM for installing AIX updates and new packages over the HTTP protocol:

Network Installation Manager (NIM) supports the installation of AIX updates over the Hypertext Transfer Protocol Secure (HTTP) protocol to conform to the emerging data center policies that restrict the use of network file server (NFS).

AIX BOS installation still requires the use of the NFS version 3 protocol or the more secure NFS version 4 protocol. In addition to the installation of filesets, NIM customization processes such as script execution and copying the `file_res` directory are supported over the HTTP protocol.

The HTTP protocol provides the following advantages for NIM management:

- All communication occur over a single HTTP port. Hence, the authorization through a firewall is easier to manage.
- AIX installation steps are driven from the client's end, that is, the target system of the installation. Therefore remote access is not required for running the commands.
- NIM or any other products that currently use the client-server model of NFS can easily use HTTP.
- Able to extend the end product to support additional protocols.

NIM HTTP Service

AIX 7.2.0 ships a new service handler that provides HTTP access to NIM resources. The `nimhttp` service is defined in the `/etc/services` and the `nimhttp` daemon, which listen for requests over the 4901 port. When the `nimhttp` service is active, NIM clients attempt to access the `/etc/services` file and request customization of the scripts that are defined in the `nimhttp` service. If HTTP access fails or if the access is denied, access failover attempt to the NFS client occurs.

Enabling the nimhttp service on the NIM server

To enable the `nimhttp` service, run the following command on the NIM server:

```
#> nimconfig -h
```

Note: The `crypto` or `ssl` setting is automatically discovered.
When the `nimhttp` service is started, the service attempts to read the `httpd.conf` configuration file that is located in the default home directory of the root user. If you are using the `nimhttp` service for the first time, and if you start the `nimhttp` service without creating a configuration file, a configuration file is created and populated with default values of the `nimhttp` service.

An example of the default `httpd.conf` configuration file follows:

```
#> cat /httpd.conf
#
#
#http service defines
#
#
service.name=nimhttp
# Designates the service name used  when discovering the listening port for requests (i.e., nimhttp)
#
service.log=/var/adm/ras/nimhttp.log
#Log of access attempts and equivalent responses. Also useful for debug purposes.
#
# service.proxy_port=
#Designates the service portnumber used when configured as a proxy.
#
#-------------------------------------------------------------
```

```
# http configuration
#------------------------------------------------------------
#
document_root=/export/nim/
#Designates the directory to serve files from.
#
enable_directory_listing=yes
#Allow requests for listing served files/directories under the document root.
#
enable_proxy=no
#Enable the webservice to act as a proxy server.
#
ssl.cert_authority=/ssl_nimsh/certs/root.pem
#Designates the file location of the certificate authority used for digital certificate signing.
#
ssl.pemfile=/ssl_nimsh/certs/server.pem
#Designates the file location of the PEM format file which contains both a certificate and private key.
#
```

The properties of the httpd.conf file

The `httpd.conf` file has the following properties and settings:

document_root path

Files that are not defined as resource locations can be accessed by using the HTTP protocol. These files must be located in the path setting of the `document_root`. The defined `document_root` path location cannot be modified when the `nimhttp` service is operational.

The `document_root` path might contain many directories. When you set the `enable_directory_listing` option, client requests can travel the `document_root` path. If the `enable_directory_listing` option is set to value of *no*, all files that are used during the installation must be located in the current working directory of the `document_root` path.

Secure Socket Layer (SSL) settings

The `nimhttp` service uses basic protocol handshake as the default authentication. You must provide valid paths for the certificate authority (CA) and the root certificate files for the server to enable a more secure Digest Authentication method.

The certificate authority and root Privacy Enhanced Mail (PEM) files used in the `nimhttp` service can be created by using the existing SSL management option in NIM. To create the `ssl.cert_authority` and `ssl.pemfiles` files that are used by the `nimhttp` service, run the following command on the NIM master:

```
#> nimconfig –c
```

Note: You do not need to create the `ssl.cert_authority` and `ssl.pemfiles` files if these SSL files exist in the current directory.

To check whether the NIM master has been configured for SSL authority before starting the `nimhttp` service by using the SSL option, run the following command on the NIM master:

```
#> lsnim –a ssl_support
```

Proxy settings

The NIM client commands depend on the `nimhttp` service because the NIM server acts as the file server that hosts the NIM resources.

Alternatively you can use the proxy option for handling an HTTP request by using the nimhttp server code. When the proxy option is enabled by using the value enable_proxy=yes, any requests for service over the nimhttp port are forwarded to the service port listed in the service.proxy_port list of ports.

The HTTP authentication is handled by the destination service and not by the nimhttp service. The destination service port is identified locally in the NIM client.

Disabling the nimhttp service on the NIM server

To disable the nimhttp service, run the following command on the NIM server:

```
#> nimconfig -H
```

NIM resources that support HTTP access

The following resources can access the http file by using the nimhttp service:

- file_res
- fix_bundle
- installp_bundle
- lpp_source
- script

Examples

1. To install the bos.sysmgt.nim.master fileset and to define basic resources, run the following command:

   ```
   #> nim_master_setup –a device=/dev/cd0
   ```

2. To enable Secure Socket Layer (SSL) management for the NIM environment, run the following command:

   ```
   #> nimconfig –c
   ```

3. To enable the nimhttp service with SSL support, run the following command:

   ```
   #> nimconfig -h
   ```

4. To check the service log file for any errors that have occurred, run the following command:

   ```
   #> cat /var/adm/ras/nimhttp.log
   ```

5. If you are using the push operation, the following commands support the nimhttp service:

   ```
   nim –o cust–a file_res= <obj_name> <client_obj_name>
   nim –o cust–a script= <obj_name> <client_obj_name>
   nim –o cust –a lpp_source=<obj_name> -a filesets=<fileset names to install>
                                                <client_obj_name>
   nim –o cust –a lpp_source=<obj_name> -a installp_bundle=<obj_name> <client_obj_name>
   nim –o cust –a lpp_source=<obj_name> -a fixes=update_all <client_obj_name>
   ```

6. If you are using the pull operation from the NIM client, the following commands support the nimhttp service:

   ```
   nimclient –o cust –a file_res=<obj_name>
   nimclient –o cust –a script=<obj_name>
   nimclient –o cust –a lpp_source=<obj_name> ---a filesets=<fileset names to install>
   nimclient –o cust –a lpp_source=<obj_name> ---a installp_bundle=<obj_name>
   nimclient –o cust –a lpp_source=<obj_name> ---a fixes=update_all
   ```

Debugging session for the nimhttp service

The communication from the NIM master to the client occurs over the nimsh protocol, to run an application. But, the client requests the file resources be sent over the HTTP protocol instead of the usual NFS export or mount process. The following steps show an example debug session.

1. To start the nimhttp service from the NIM master, run the following command:
   ```
   nimconfig -h
   ```
2. To keep the current window active from the NIM master for viewing the HTTP requests from the client, run the following command.
   ```
   tail -f /var/adm/ras/nimhttp.log
   ```
3. In a separate window, either on the client or master system, run the cust operation that you want from a system on which AIX 7.2 is installed.
4. The log activity for the nimhttp service is displayed on the terminal window.

Confirming the use of HTTP instead of NFS

To ensure that the NIM cust operations are performed by using the HTTP protocol and not by using the NFS protocol, ensure that the NFS cannot access the NIM resources by removing entries from the /etc/exports file. For instructions, see steps 1 - 4 in the Debugging session for the nimhttp service section. After the NIM cust operation starts downloading filesets, run the exportfs -uav command to ensure that the NIM master does not failover to an NFS mount from the client.

If a nimhttp service request is received successfully, a log entry similar to the following example is displayed:

```
------
Mon Oct 26 14:45:37 2015
nim_http: data string passed to get_http_request: "GET /client.defs HTTP/1.1

Connection: close

"

Mon Oct 26 14:45:37 2015 Request Type is GET
Mon Oct 26 14:45:37 2015 Sending Response Header "200 OK"

Mon Oct 26 14:45:37 2015 Sending file over socket 5. Expected length is 2989
Mon Oct 26 14:45:37 2015 Total length sent is 2989

Mon Oct 26 14:45:37 2015 handle_httpGET: Entering cleanup statement
```

Verifying the NIM environment configuration

You can use the following steps to confirm whether the NIM environment is configured properly for handling nimhttp services.

1. To verify whether the NIM master is listening for connection requests over a specific host address, run the following command on the NIM master:
   ```
   #> cat /etc/niminfo

   #> nimconfig -h (if necessary)

   #> netstat -a | grep nimhttp
   #> netstat -i
   #> cat /httpd.conf
   #> cat /var/adm/ras/nimhttp.log
   ```
   On the client, run the commands:
   ```
   #> cat /etc/niminfo
   #> cd /tmp
   #> nimhttp b f /export/nim b o dest=/tmp –v
   ```
   To determine whether the client request has reached the NIM master, run the following command on the NIM master:
   ```
   #> cat /var/adm/ras/nimhttp.log
   ```

2. If the commands in step 1 result in an unexpected output, the client might be requesting the `nimhttp` service from a host IP on which the NIM master does not respond. You can check the list of the host name and IP addresses on which the master system is running. You can provide the host name as an argument to the `nimhttp` command that is provided during the previous client request. You can run the `nimhttp -?` command to understand the flag syntax of the `nimhttp` command.

Adding a management object to the NIM environment

You can add management objects for stand-alone, diskless, and dataless clients to the NIM environment with SMIT or by using the command line.

You can add management objects to the NIM environment to gain additional control on the standard NIM machine objects.

The stand-alone, diskless, and dataless clients can be either *managed* or *unmanaged*. A *managed* client is associated with a managing system that controls the client.

Managed clients can use the network-boot and power-control capabilities of the **dsm.core** fileset when the fileset is installed. For example, it is possible to request a maintenance boot of the client without accessing the managed system to request a network boot.

When the **dsm.core** fileset is installed, the additional capabilities of *managed* clients, in comparison to the *unmanaged* clients, is as follows:

- Performs a network boot and boot in maintenance mode by using the following command:
  ```
  nim -o maint_boot -a boot_client=yes
  ```
- Performs a network boot and the client is installed by using the following command:
  ```
  nim -o bos_inst -a boot_client=yes
  ```
- Boots or reboots the client by using the following command:
  ```
  nim -o reboot
  ```
- Opens a virtual console **xterm** when you use the **-a open_console** parameter on selected NIM operations.
- Defines and uses virtual optical devices so that **VIOS** clients can mount ISO images from a virtual CD.

Management objects are represented by Hardware Management Console (HMC), Central Electronic Complex (CEC), Integrated Virtualization Manager (IVM), Virtual I/O Server (VIOS), Blade Center Management Module (BCMM), or Power® Virtualization Center (PowerVC) objects.

Adding HMC management objects to the NIM environment:

Follow the instructions to add a Hardware Management Console management object.

HMC object represents a Hardware Management Console (HMC) system. To add an HMC object, the operation requires the **dsm.core** fileset to be installed on the NIM master.

To add an HMC object from the command line, follow these steps:

1. Create an encrypted password file that contains the login ID and related password on the NIM master to access the HMC. It must be created using the **dpasswd** command from the **dsm.core** fileset. If you do not want the password displayed in clear text, exclude the **-P** parameter and the **dpasswd** command will prompt for the password.
   ```
   # dpasswd -f EncryptedPasswordFilePath -U hmcLogin -P hmcPassword
   ```
2. Pass the encrypted password file in the **passwd_file** attribute by using the **define** command of the HMC.

```
# nim -o define -t hmc -a passwd_file=EncryptedPasswordFilePath \
-a if1=InterfaceDescription \
-a net_definition=DefinitionName \
HMCName
```

3. If the network object that describes the network mask and the gateway used by HMC does not exist, use the **net_definition** attribute. After you remove the HMC objects, the file specified by the **passwd_file** attribute must be removed manually.

Example

To add an HMC object with the host name **hmc1** that has the following configuration:

```
host name=hmc1
password file path=/etc/ibm/sysmgt/dsm/config/hmc1
network type=ethernet
subnet mask=255.255..240.0
default gateway=gw1
default gateway used by NIM master=gw_maste
```

Enter the following command:

```
# nim -o define -t hmc -a passwd_file=/etc/ibm/sysmgt/dsm/config/hmc1 \
-a if1="find_net hmc1 0" \
-a net_definition="ent 255.255.240.0 gw1 gw_master" hmc1
```

For additional information, see**/opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf** of the **dsm.core** fileset.

Adding CEC management objects to the NIM environment:

Follow the instructions to add a Central Electronic Complex management object.

A Central Electronic Complex (CEC) object is managed by a Hardware Management Console (HMC), which requires the **dsm.core** fileset to be installed on the NIM master.

To define a CEC object, the CEC type, model, and serial number must be retrieved. You can use one of the following method to obtain the required information.

Note: The following methods describe the procedure to add a CEC object from the command line:

- **Method 1**: Use the **nimquery** command to retrieve information and define the CEC objects. If the HMC object is defined in NIM and the **openssh.base.client** fileset is installed, you can use the **nimquery** command to define the CEC managed by a given HMC. On the NIM master, type the following command:

  ```
  # nimquery -a hmc=hcmObjectName -d
  ```

  The name of each NIM CEC object that is defined by the **nimquery** command is in the form: *cec_type cec_model_cec_serial_number*
- **Method 2**: Use the **nimquery** command to retrieve the information that is required to define the CEC object. On the NIM master, type:

  ```
  # nimquery -a hmc=hcmObjectName -p
  ```

  The CEC object is defined on the output.
- **Method 3**: Log in to the HMC object and use the **lssyscfg** command to retrieve the CEC information. On the NIM master, type:

  ```
  # ssh hmcLogin@hmcHost lssyscfg -r sys -F name,type_model,serial_num
  ```

Defining CEC object

When you use any of these methods to define a CEC object, follow these steps:

For Method 1, the CEC object is defined by the **nimquery** command.

For Method 2 and Method 3, retrieve the information from the CEC and type the following command on the NIM master:

```
# nim -o define -t cec -a hw_serial=cecSerialNumber \
-a hw_type=cecType -a hw_model=cecModel \
-a mgmt_source=hmcObject cecName
```

Example

To add the CEC object with HMC name **hmc1** that has the following configuration:

```
cec object name=cec1
hmc object name=hmc1
cec type=9115
cec model=505
cec serial number=10069DA
```

Enter the following command:

```
# nim -o define -t cec -a hw_serial=10069DA \
-a hw_type=9115 -a hw_model=505 \
-a mgmt_source=hmc1 cec1
```

Adding VIOS management objects to the NIM environment:

Follow the instructions to add a Virtual I/O Server management object.

A Virtual Input or Output Server (VIOS) object is managed by a CEC object in NIM. These operations require the **dsm.core** fileset to be installed on the NIM master.

To add VIOS object from the command line, follow these steps:

To define a VIOS object, the VIOSLPAR identifier must be retrieved. You can use one of the method to obtain the information.

1. **Method 1**: Use the **nimquery** command to retrieve the information. Define the CEC object in NIM and if the **openssh.base.client** fileset is installed, use the **nimquery** command to retrieve the VIOS attributes. On the NIM master, type:

   ```
   # nimquery -a cec=cecObjectName -p
   ```

   Part of the information about each LPAR of the CEC object including the Virtual I/O Server is displayed.
2. **Method 2**: Log into the HMC object and use the **lssyscfg** command to retrieve the VIOS information. For example, to use the following configuration:

   ```
   cec name on the hmc = cec1
   HMC login = hmcLogin
   HMC host name = hmcHost
   ```

   On the NIM master, type:

   ```
   # ssh hmcLogin@hmcHost lssyscfg -r lpar -m cec1 -F name,lpar_type,lpar_id | grep vioserver
   ndaf1ios_lpar,vioserver,2
   ```
3. If the network object that describes the network mask and the gateway used by the IVM does not exist, use the **net_definition** attribute.

Note: You must manually remove the file pointed to the **passwd_file** attribute after you remove the IVM objects.

Defining the VIOS object

On the NIM master, type the following after you have retrieved the VIOS object:

```
# nim -o define -t vios -a platform=PlatformType \
-a netboot_kernel=NetbootKernelType \
-a if1=InterfaceDescription \
-a mgmt_source=cecObjectName -a identity=viosLPARIdentifier \
-a net_definition=DefinitionName -a ring_speed1=SpeedValue \
-a cable_type1=TypeValue -a iplrom_emu=DeviceName ViosName
```

Example

To add a machine with the host name **machine1** that has the following configuration:

```
host name=vios1
cec object name=cec1
vios lpar identifier=2
platform=chrp
kernel=64
network type=ethernet
subnet mask=255.255.240.0
default gateway=gw1
default gateway used by NIM master=gw_master
cable type=N/A
network boot capability=yes (no emulation needed)
```

Enter the following command sequence:

```
# nim -o define -t vios -a platform="chrp" \
-a netboot_kernel="64" -a if1="find_net vios1 0" \
-a cable_type1="N/A" \
-a mgmt_source=cec1 -a identity=2 \
-a net_definition="ent 255.255.240.0 gw1 gw_master" machine1
```

For additional information, see the file **/opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf** of the **dsm.core** fileset.

Adding IVM management objects to the NIM environment:

Follow the instructions to add an integrated virtualization manager management object.

An IVM represents an Integrated Virtualization Management (IVM) logical partition (LPAR). These operations require the **dsm.core** fileset to be installed on the NIM master.

To add an IVM object from the command line, follow these steps:

1. Create an encrypted password file that contains the login ID and related password to access the IVM object on the NIM master by using the **dpasswd** command from the **dsm.core** fileset. If you do not want the password to display in clear text, exclude the **-P** parameter. The **dpasswd** command will then prompt for the password.

   ```
   # dpasswd -f EncryptedPasswordFilePath -U ivmLogin -P ivmPassword
   ```

2. Pass the encrypted password file that is created in the **passwd_file** attribute by using the **define** command of the IVM object:

   ```
   # nim -o define -t ivm -a passwd_file=EncryptedPasswordFilePath \
   -a if1=InterfaceDescription \
   -a net_definition=DefinitionName \
   ivmName
   ```

3. If the network object describing the network mask and the gateway used by the IVM object does not exist, use the **net_definition** attribute.

   Note: You must manually remove the file pointed to the **passwd_file** attribute after you remove the IVM objects.

Example

To add the IVM object with the host name **ivm1**, that has the following configuration:

```
host name=ivm1
password file path=/etc/ibm/sysmgt/dsm/config/ivm1
network type=ethernet
subnet mask=255.255.240.0
default gateway=gw1
default gateway used by NIM master=gw_maste
```

Enter the following command sequence:

```
# nim -o define -t ivm -a passwd_file=/etc/ibm/sysmgt/dsm/config/ivm1 \
-a if1="find_net ivm1 0" \
-a net_definition="ent 255.255.240.0 gw1 gw_master" ivm1
```

For additional information, see the**/opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf** file of the **dsm.core** fileset.

Adding BCMM management objects to the NIM environment:

Follow the instructions to add a blade center management module management object.

A BCMM object represents a Blade Center Management Module (BCMM). These operations require the **dsm.core** fileset to be installed on the NIM master.

To add a BCMM object from the command line, follow these steps:

1. Create an encrypted password file that contains the login ID and related password to access the BCMM object. The BCMM object is accessed on the NIM master by using the **dpasswd** command from the **dsm.core** fileset. If you do not want the password to be displayed in clear text, exclude the **-P** parameter. The **dpasswd** command prompts for the password.

   ```
   # dpasswd -f EncryptedPasswordFilePath -U bcmmLogin -P bcmmPassword
   ```

2. Pass the encrypted password file in the **passwd_file** attribute by using the **define** command of the BCMM as follows:

   ```
   # nim -o define -t bcmm -a passwd_file=EncryptedPasswordFilePath \
   -a if1=InterfaceDescription \
   -a net_definition=DefinitionName \
   bcmmName
   ```

3. If the network object that describes the network mask and the gateway used by the BCMM object does not exist, use the **net_definition** attribute.

Note: The file pointed to the **passwd_file** attribute must be manually removed when you remove the BCMM objects.

Example

To add the BCMM object with host name **bcmm1** that has the following configuration:

```
host name=bcmm1
password file path=/etc/ibm/sysmgt/dsm/config/bcmm1
network type=ethernet
subnet mask=255.255..240.0
default gateway=gw1
default gateway used by NIM master=gw_maste
```

Enter the following command sequence:

```
# nim -o define -t bcmm -a passwd_file=/etc/ibm/sysmgt/dsm/config/bcmm1 \
-a if1="find_net bcmm1 0" \
-a net_definition="ent 255.255.240.0 gw1 gw_master" bcmm1
```

For additional information, see **/opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf** of the **dsm.core** fileset.

Adding a nas_filer management object to the NIM environment:

Follow the instructions to add a nas_filer management object.

If you define resources on a network-attached storage (NAS) device by using the nas_filer management object, you can use those resources without changing the network information and configuration definition changes on the Shared Product Object Tree (SPOT) server. To add a nas_filer object, the **dsm.core** fileset must be installed on the NIM master.

To add a nas_filer object from the command line, complete the following steps:

1. Create an encrypted password file that contains the login ID and related password on the NIM master to access the nas_filer object. The encrypted password file must be created by using the **dpasswd** command from the **dsm.core** fileset. If you do not want the password to be displayed in clear text, exclude the **-P** parameter. The **dpasswd** command prompts for the password. Use the following command as an example:

   ```
   # dpasswd -f EncryptedPasswordFilePath -U nas_filerLogin -P nas_filerPassword
   ```

2. Pass the encrypted password file in the **passwd_file** attribute by using the **define** command of the nas_filer object. Use the following command as an example:

   ```
   # nim -o define -t nas_filer -a passwd_file=EncryptedPasswordFilePath \
   -a if1=InterfaceDescription \
   -a net_definition=DefinitionName \
   nas_filerName
   ```

3. If the network object that describes the network mask and the gateway that is used by the nas_filer object does not exist, use the **net_definition** attribute. After you remove the nas_filer objects, the file that is specified by the **passwd_file** attribute must be removed manually.

Example

To add a nas_filer object that has the host name **nf1** and the following configuration:

host name=nf1
password file path=/etc/ibm/sysmgt/dsm/config/nf1
network type=ethernet
subnet mask=255.255.240.0
default gateway=gw1
default gateway used by NIM master=gw_maste, enter the following command:

```
# nim -o define -t nas_filer -a passwd_file=/etc/ibm/sysmgt/dsm/config/nf1 \
-a if1="find_net nf1 0" \
-a net_definition="ent 255.255.240.0 gw1 gw_master" nf1
```

For more information about adding a nas_filer object, see the technical note that is included in the **dsm.core** fileset (**/opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf**).

Adding PowerVC management objects to the NIM environment:

You can add an IBM Power Virtualization Center (PowerVC) management object to the NIM environment.

A PowerVC object represents the PowerVC management server that is used for system management operations. The dsm.core fileset must be installed on the NIM master to perform PowerVC operations.

To add a PowerVC object from the command line, complete the following steps:

1. Create an encrypted password file that contains the login ID and related password on the NIM master to access the PowerVC object. The encrypted password file must be created by using the **dpasswd** command from the dsm.core fileset as shown in the following example:

   ```
   # dpasswd -f EncryptedPasswordFilePath -U powervcLogin -P powervcPassword
   ```

   Note: If you do not want the password to be displayed in clear text, exclude the **-P** flag. When you do not specify the **-P** flag, the **dpasswd** command prompts for the password.
2. Specify the encrypted password file with the **passwd_file** attribute by using the **define** command of the PowerVC object as shown in the following example:

   ```
   # nim -o define -t powervc -a passwd_file=EncryptedPasswordFilePath \
   -a if1=InterfaceDescription \
   -a net_definition=DefinitionName \
   powervcName
   ```
3. If the network object, which describes the network mask and the gateway that is used by the PowerVC object, does not exist, specify the **net_definition** attribute. After you remove the PowerVC objects, manually remove the file that is specified by the **passwd_file** attribute.

Example

To add a PowerVC object that has the following configuration setting:

```
host name=pvc1
password file path=/etc/ibm/sysmgt/dsm/config/pvc1
network type=ethernet
subnet mask=255.255.240.0
default gateway=gw1
default gateway used by NIM master=gw_master
```

Enter the following command:

```
# nim -o define -t powervc  \
-a passwd_file=/etc/ibm/sysmgt/dsm/config/pvc1 \
-a if1="find_net nf1 0" \
-a net_definition="ent 255.255.240.0 gw1 gw_master" pvc1
```

For more information about adding a PowerVC object, see the technical note that is included in the dsm.core fileset (/opt/ibm/sysmgt/dsm/doc/dsm_tech_note.pdf).

Adding standalone clients to the NIM environment

You can add standalone clients to the NIM environment with SMIT, or the command line.

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. You can add a client with or without the network information.

Adding a client with the network information using SMIT:

Follow these instructions to add a standalone NIM client to the NIM environment using SMIT.

To add a standalone NIM client to the NIM environment using SMIT, use Method A if the client machine is not running or if the client does not have AIX 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 installation from a **force-push** operation. This procedure automatically adds NIM networks when needed.

To add a standalone NIM client that already has AIX installed, use Method B.

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

- The NIM master must be configured. For more information, see "Configuring the NIM master and creating basic installation resources" on page 121.
- You must know the subnet mask, the default gateway for the client machine, and the default gateway for the NIM master.

Adding a client with the network information using SMIT when client is not running (method A):

Follow these steps to add a client with the network information using SMIT when the client is not running.

1. On the NIM master, add a standalone client to the NIM environment by typing 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.

Adding a client with the network information using SMIT when client is running (method B):

Follow these steps to add a client with the network information using SMIT when the client is running.

1. On a system that you have chosen to be a NIM client, verify that if the `bos.sysmgt.nim.client` fileset is installed by typing the following: `# lslpp -L bos.sysmgt.nim.client`
2. If the `bos.sysmgt.nim.client` fileset is not installed, then install the fileset from the *AIX Volume 1* CD/DVD by typing the following: `# installp -acXd /dev/cd0 bos.sysmgt.nim.client`
3. Enter the `smit niminit` fast path.
4. 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.

Adding a client with the network information from the command line:

Follow these instructions to add a standalone NIM client to the NIM environment from the command line.

To add a standalone NIM client to the NIM environment from the command line, use Method A if the client machine is not running or if the client does not have AIX 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 installation from a **force-push** operation. This procedure automatically adds NIM networks when needed.

To add a standalone NIM client that already has AIX installed, use Method B.

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

- The NIM master must be configured. For more information, see "Configuring the NIM master and creating basic installation resources" on page 121.
- You must know the subnet mask, the default gateway for the client machine, and the default gateway for the NIM master.

Adding a client with the network information from the command line when client machine is not running (method A):

Follow these steps to add a client with the network information from the command line when client machine is not running.

On the NIM master, type:

```
# 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=chrp
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="chrp" \
-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=chrp
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
```

enter the following command sequence:

```
# nim -o define -t standalone -a platform="chrp" \
-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" machine2
```

Note:

1. 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.

2. For more information about the attributes you can specify when defining NIM clients, see "Defining NIM clients" on page 108

Adding a client with the network information from the command line when client machine is running (method B):

Follow these steps to add a client with the network information from the command line when the client machine is running.

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=chrp
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=chrp -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=chrp
netboot_kernel=up
ring speed1=16
```

enter the following command sequence:

```
# niminit -a name=machine2 -a master=master_mac \
-a pif_name=tr0 -a platform=chrp -a netboot_kernel=up \
-a ring_speed1=16
```

Adding a client without the network information using SMIT:

You can use a new remote service for defining clients in the NIM environment. Follow these directions to use this new service with SMIT.

The new service is called the NIM Service Handler (NIMSH), and it runs on potential NIM clients. When you define a system using NIMSH, no information is required for defining the client object. For additional information, see "Using the NIM service handler for client communication" on page 156.

To define NIM clients using **nimquery**, complete the following steps:

1. Type the fast path `smitty nim_query` on the NIM master.

2. Specify the hostname of the machine to query.

   Note: The machines must have NIMSH daemon active.
3. Select **yes** as the option for **Adding Machine to the NIM Environment** if you are adding the machine as a NIM client object.
4. Specify the new client object name.

Adding a client without the network information from the command line:

You can use a new remote service for defining clients in the NIM environment. Follow these directions to use the new service from the command line.

The new service is called the NIM Service Handler (NIMSH), and it runs on potential NIM clients. When you define a system using NIMSH, no information is required for defining the client object. For additional information, see "Using the NIM service handler for client communication" on page 156.

To define NIM clients using the **nimquery** command on the command line, type the following:

```
# nimquery -a host=hostname -a name=obj_name -d
```

For more information on defining NIM clients using NIMSH, see the **nimquery** command.

Verifying the status of your client machine:

Use the **niminit** command to verify the status of your client machine.

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

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

The system returns output similar to the following:

```
Standalone2:
   class          =  machines
   type           =  standalone
   Cstate         =  ready for a NIM operation
   platform       =  chrp
   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.

Adding WPAR clients to the NIM environment

You can use SMIT or the command line to add WPAR clients to the NIM environment,

Adding a WPAR client to the NIM environment using SMIT:

Use this procedure to add a WPAR client to the NIM environment using SMIT.

1. To define a workload partition client, enter the smit nim_mkmac fast path.

2. Specify the host name of the machine.
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 to define the client machine.

Adding a WPAR client to the NIM environment using the command line:

Use this information to add a WPAR NIM client to the NIM environment from the command line.

The following are prerequisites for using this procedure:

- The NIM master must be configured. For information about configuring the NIM master, see "Configuring the NIM master and creating basic installation resources" on page 121.
- You must know the subnet mask, the default gateway for the client machine, and the default gateway for the NIM master.

To define a WPAR client, enter the following:

```
 -a mgmt_profile1=ManagingSystemDescription \
-a if1=InterfaceDescription \
<optional resources and attributes>\
MachineName
```

For example, the command to add the wpar1 WPAR client that is managed by the nim_std1 NIM stand-alone client to the NIM environment is as follows:

```
nim -o define -t wpar -a mgmt_profile1="nim_std1 wpar1" \
-a if1="find_net wpar1 0" wpar1
```

For detailed attribute information, see "Diskless and dataless clients" on page 113.

Using NIM with ATM networks

Special processing is required to install a machine over an ATM network.

Unlike other network adapters, ATM adapters cannot be used to boot a machine. Installing a machine over an ATM network requires special processing. 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.

Because 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. The NIM **bos_inst** operation must copy a boot image to the hard disk of the client before the machine is rebooted. Some Object Data Manager (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 programs installed to support the special processing required for installation over ATM, so the `/usr/lib/boot/bin` and `/usr/lpp/bos.sysmgt/nim/methods` directories are mounted at the client from the NIM master. These directories contain the programs that 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 because 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.

Converting a generic network into an ATM network:

You can convert generic networks into ATM networks.

Prerequisites

- Machines that will have the 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.

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.

To change the name of the network, type the following:

```
nim -o change -a new_name=new_network_name current_network_name
```

Recovering a client on an ATM network after boot failure:

Follow this procedure for recovering a client on an ATM network after boot failure.

Because 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. If the installation is stopped 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. Use the following procedure:

1. Boot the client from the CD/DVD.
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`

Stopping the reboot of a client on an ATM network:

Follow this procedure for stopping the reboot of a client on an ATM network.

If errors are detected during the NIM **bos_inst** operation and the client machine has not 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 field in the output is the name of the job. For example:

```
$ at -1
root.884205595.a Wed Jan   7 14:39:55 1998
```

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

   For example:

   ```
   $ 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 typing:

```
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

- 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.
- If the software is to be installed on a **SPOT** resource, the server of the **SPOT** must be running.
- 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**.)

Customizing NIM clients and SPOT resources by using SMIT:

Follow this procedure for customizing NIM clients and SPOT resources using SMIT.

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.

Note: If you select the **Invoke live update?** option, the AIX Live Update operation is run against the `TARGET` client. The `TARGET` client must be a Network Installation Manager (NIM) standalone system. If the **LIVE_UPDATE_DATA** option is selected with a `live_update_data` NIM resource, you can NFS-export the resource to the client and the resource is used for Live Update. If a `live_update_data` resource is not specified for the **LIVE_UPDATE_DATA** field, the file at the `/var/adm/ras/liveupdate/lvupdate.data` location on the client is used instead for the Live Update operation.

Customizing NIM clients and SPOT resources from the command line:

Follow this procedure for customizing NIM clients and SPOT resources 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
```

Example 3:

To run a Live Update operation against a client `machA`, by using the **live_update_data** resource, `liveupdate_machA`, with an interim fix of `IY12345` that uses the **lpp_source** resource named `lpp_source1`, enter:

```
nim -o cust -a live_update=yes -a live_update_data=liveupdate_machA \
-a lpp_source=lpp_source1 -a filesets="IY12345" machA
```

Example 4:

To run a Live Update operation in preview mode against a client `machA`, by using the **live_update_data** resource, `liveupdate_machA`, with an interim fix of `IY12345` that uses the **lpp_source** resource, `lpp_source1`, enter:

```
nim -o cust -a live_update=yes -a live_update_data=liveupdate_machA -a installp_flags="-p" \
-a lpp_source=lpp_source1 -a filesets="IY12345" machA
```

Note: Several other resources and attributes can be specified on the command line with the **cust** operation. For a complete description of the **cust** operation, see "Using NIM operations" on page 257.

Installing an Interim Fix into a SPOT resource:

Follow this procedure for installing an interim ifix into a SPOT resource, or to patch a shared operating system file or concurrent update of a thin server to disk.

Use the following procedure to install an interim fix into a NIM SPOT resource.

The interim fix is usually named *<Label>.<Timestamp>.epkg.Z*

Installation to a NIM SPOT

1. 1. Check if an APAR containing the desired fix is installed on the NIM master and SPOT: For example:
   - AIX 5.1: APAR IY40088

- AIX 5.2: APAR IY40236

To check if the APAR is installed on the NIM master, type:

```
# instfix -ik <APAR>
```

To check if the APAR is installed on the NIM SPOT, type:

```
 # nim -o fix_query -a fixes=<APAR><Spot_Name>
```

2. Create an interim fix path in any lpp_source (if it does not exist already). The path will be in the format of: *lpp source path>emgr/ppc*

   Example:

   ```
   # lsnim -a location 520lpp
   520lpp:
   location = /520/520lpp

   # mkdir -p /520/520lpp/emgr/ppc
   ```

3. Copy the ifix package to the ifix path in the lpp_source:

   ```
   cp <EFix_File><LPP_Location>/emgr/ppc
   ```

   Example:

   ```
   # cp IY12345.050303.epkg.Z /520/520lpp/emgr/ppc
   ```

4. Execute a nim "cust" operation on the SPOT specifying the LPP_SOURCE and the interim fix:

   ```
    # nim -o cust -a lpp_source=<LPP_Source>-a filesets=<Interim fix><Spot>
   ```

   Example

   ```
   # nim -o cust -a lpp_source=520lpp -a filesets=IY12345.050303.epkg.Z 520spot
   ```

List Interim Fixes Installed in a SPOT

To list out all interim fixes installed in a SPOT, use the lslpp nim query with an lslpp tag of e:

```
# nim -o lslpp -a lslpp_flags=e<Spot>
```

Example

```
# nim -o lslpp -a lslpp_flags=e 520spot
D     STATE    LABEL           INSTALL TIME                ABSTRACT
=== =====  ========== ================== ====================
1        S        IY12345         08/13/04 13:19:20           IY12345 AIX 5.2 efix
```

Uninstall Interim Fix from a SPOT

To uninstall the ifix from the SPOT, use the fix <*Label*> with a maint command on the SPOT (note: the label is related to, but not exactly the filename; it is the first part of the filename):

```
 nim -Fo maint -a installp_flags=u -a filesets=<Label><Spot_Name>
```

Example

```
  # nim -Fo maint -a installp_flags=u -a filesets=IY12345 520spot
```

Note: Installing this interim fix will lock the affected fileset to prevent the installation of an update not containing the fix from regressing the system. Once the official fix is available, you may use the interim fix uninstall command to uninstall the ifix before applying the official APAR.

Listing interim fixes installed in a SPOT:

Basic instructions list all interim fixes installed in a SPOT.

To list all interim fixes installed in a SPOT, use the lslpp NIM query with the lslpp flag e:

```
# nim -o lslpp -a lslpp_flags=e <Spot>
```

Example

```
# nim -o lslpp -a lslpp_flags=e 520spot
ID  STATE   LABEL     INSTALL      TIME      ABSTRACT
=== =====  ========== ==================     ====================
1    S      IY12345   08/13/04 13:19:20   IY12345   AIX 5.2 efix
```

Uninstalling an interim fix from a SPOT:

Use the instructions to uninstall an interim fix from a SPOT.

To uninstall an interim fix from the SPOT, use the fix <*Label*> with a maint command on the SPOT. The label is related to, but not exactly the filename. It is the first part of the filename.

```
nim -Fo maint -a installp_flags=u -a filesets=<Label> <Spot_Name>
```

Example:

```
# nim -Fo maint -a installp_flags=u -a filesets=IY12345 520spot
```

Note: Installing this interim fix locks the affected fileset to prevent the installation of an update that does not contain the fix from regressing the system. After the official fix is available, you can use the interim fix deinstall command to uninstall the interim fix before applying the official APAR.

Configuring the NIM master and creating resources to support diskless and dataless clients

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 121.

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 300 MB of available disk space. If such space is not available, see "Using client machines as resource servers" on page 163, and "Defining an lpp_source on DVD-ROM versus hard disk" on page 151.

Configuring the NIM master and creating resources to support diskless and dataless clients using SMIT:

Follow this procedure for configuring the NIM master and creating resources to support diskless and dataless clients using SMIT.

1. Insert the AIX media into the media 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, because 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**.

Configuring the NIM master and creating resources to support diskless and dataless clients from the command line:

Follow this procedure for configuring the NIM master and creating resources to support diskless and dataless clients from the command line.

1. Insert the AIX media into the media 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, type: `mkdir /cdfs`.
3. To create a `cdrom` file system, type: `crfs -v cdrfs -p ro -d'cd0' -m'/cdfs'`
4. To mount the disk, type: `mount /cdfs`
5. To install the `bos.sysmgt.nim` fileset from the disk, type: `installp -agX -d /cdfs/usr/sys/inst.images bos.sysmgt.nim`.

   or to install the `bos.sysmgt.nim` fileset from a tape, type: `installp -agX -d /dev/rmt0 bos.sysmgt.nim`
6. If installing from CD/DVD, to unmount the `cdrom` file system, type: `unmount /cdfs`
7. To configure the NIM master using the **nimconfig** command, type:

   ```
   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 = chrp
   kernel type = mp
   ```

   enter the following command sequence:

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

Note: For additional attribute information, see the **nimconfig** command.

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 jfs2 -g rootvg -a size=$((2000*200)) \
-m /export/spot -A yes -p rw -t 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 jfs2 -g rootvg -a size=$((2000*150)) \
-m /export/dd_resource -A yes -p rw -t 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, which is required for diskless and dataless clients unless a shared_root resource (for diskless clients only) is used, enter:

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

    To create the shared_root resource named shared_root1, which is required for diskless clients unless a shared_root resource is used, enter:

```
# nim -o define -t shared_root -a server=master \
-a location=/export/dd_resource/shared_root1 \
-a spot=spot1 shared_root1
```

    To create the dump resource named dump1 (optional), 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:

    a. The file systems created for the NIM resources are not required, but they can be beneficial for storage management.

b. For more information about NIM resources, see “Using NIM resources” on page 224.

Adding a diskless or dataless client to the NIM environment

Use this procedure 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 193 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

- 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” on page 144.
- You must know the subnet mask, the default gateway for the client machine, and the default gateway for the NIM master.

Adding a diskless or dataless client to the NIM environment using SMIT:

Follow this procedure for adding a diskless or dataless client to the NIM environment using 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.

Adding a diskless or dataless client to the NIM environment from the command line:

Follow this procedure for adding a diskless or dataless client to the NIM environment 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 108.

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
```

Note: 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.

Uninitializing diskless and dataless machines

Diskless and dataless machines are uninitialized by performing the **reset** operation.

The **reset** operation 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.

Uninitializing diskless and dataless machines using SMIT:

Follow this procedure for uninitializing diskless and dataless machines using 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**.

Uninitializing diskless and dataless machines from the command line:

Follow this procedure for uninitializing diskless and dataless machines 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
   ```

Tuning client-request processing

For large installation environments, NIM can be scaled to support anywhere from 20 to 150 client requests simultaneously. NIM scaling is done by enabling the multithreaded option on the **nimesis** daemon.

The multithreaded option provides better handling of the volume of client information change requests and client state changes. Without the use of the multithreaded option, the NIM master can become overloaded by activity on the NIM database and the number of active processes, resulting in simultaneous failures during the installation of a large number of client machines.

The multithreaded **nimesis** daemon will serialize and buffer NIM client requests to protect the NIM master from process overload, 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 changes for any client, however, are eventually processed. Debugging of 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. Because most of the NIM client 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 can support two to four clients that are installing BOS at the same time. For example, when installing 150 machines, 50 to 75 threads is 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 multithreaded daemon can monopolize system resources on the master that will not be used. For example, when installing 50 machines simultaneously, 20 to 25 threads or even the single-threaded daemon would suffice.

Note: The multithreaded option alone will not allow more machines to be installed simultaneously. The multithreaded 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.

Tuning client-request processing using SMIT:

You can tune client-request processing from the SMIT interface.

Type the SMIT fast path:

```
smit nim_tune_nimesis
```

Tuning client-request processing from the command line:

You can tune client-request processing from the command line.

The **max_nimesis_threads** attribute can be used to tune client-request processing. To enable the multithreaded **nimesis** daemon, set a value to the **max_nimesis_threads** attribute 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 multithreaded **nimesis** daemon, set a null value to the **max_nimesis_threads** attribute on the NIM master:

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

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.

Unconfiguring the NIM master using SMIT:

Follow this procedure for unconfiguring the NIM master using SMIT.

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.

Unconfiguring the NIM master from the command line:

Follow this procedure for unconfiguring the NIM master from the command line.

Enter **nim -o unconfig master**.

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 or maintenance on the machine.

Defining /usr versus non-/usr SPOTs using SMIT:

Follow this procedure for defining /usr versus non-/usr SPOTs using 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.

Creating the /usr-SPOT from the command line.:

Follow this procedure for creating the **/usr-SPOT** from the command line.

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:

After 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 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.

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, 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 DVD-ROM versus hard disk

You can define an lpp_source on a CD-ROM versus a disk using the SMIT, or the command line.

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, you can use a directory mounted from DVD-ROM installation media as the **lpp_source**.

Defining an lpp_source on CD/DVD-ROM versus hard disk using SMIT:

Follow this procedure for defining an lpp_source on CD/DVD-ROM versus disk using SMIT.

1. Mount the CD/DVD 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/DVD-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.

Defining an lpp_source on CD/DVD-ROM versus hard disk from the command line:

Follow this procedure for defining an lpp_source on a CD/DVD-ROM versus a disk from the command line.

1. Mount the CD/DVD 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/DVD 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
   ```

Using secondary adapters

Previously, during a NIM **rte** BOS installation operation, only the network adapter and interface used during BOS installation were configured. Using NIM secondary adapter definitions, you can have additional network adapters and interfaces configured during a BOS installation or customized installation.

The **nimadapters** command parses a secondary adapter stanza file to build the files required to add NIM secondary adapter definitions to the NIM environment as part of an **adapter_def** resource. The **nimadapters** command does not configure secondary adapters. The configuration takes place during a **nim -o bos_inst** operation or a **nim -o cust** operation that references the **adapter_def** resource.

Secondary adapter support is available for AIX. Before you enable a secondary adapter, you must verify the AIX version the client is running. The secondary adapters will fail to configure, because NIM is unable to find the **/usr/lpp/bos.sysmgt/nim/methods/c_cfgadptrs** client method. The following example shows the outcome if you attempt to enable this support on your NIM master.

```
nim -o cust -a adapter_def=adapter_def1 rspc10
trigger.austin.xyz.com. 0042-001 nim: processing error encountered on "master":
0042-001 m_cust: processing error encountered on "rspc10":
0042-175 c_script: An unexpected result was returned by the
"trigger.austin.xyz.com:/export/nim/scripts/rspc10.script" command:
/tmp/_nim_dir_4714/script[10]: /usr/lpp/bos.sysmgt/nim/methods/c_cfgadptrs:  not found.
```

The secondary adapter stanza file is processed by the **nimadapters** command and turned into a file that contains one stanza for each secondary adapter or interface on the NIM client. During a BOS installation, NIM processes this information and configures the secondary adapters. If a secondary adapter is already configured in the requested manner, NIM does not reconfigure the secondary adapter.

Note: Before using the **nimadapters** command, you must configure the NIM master. For information on configuring the NIM master, see "Configuring the NIM master and creating basic installation resources" on page 121.

Secondary adapter files:

This is an example a secondary adapter file.

```
 # Set default values.
 default:
     machine_type  = secondary
     subnet_mask   = 255.255.240.0
     network_type  = en
     media_speed   = 100_Full_Duplex

 # Define the machine "lab1"
 # Take all defaults and specify 2 additional attributes.
 # Unlike the case of the client definitions that are input to the
 # nimdef command, the secondary adapter definition includes at least
 # one required field that cannot be defaulted.
 lab1:
     netaddr = 9.53.153.233
     location = P2-I1/E1

 # Change the default "media_speed" attribute.

 default:
     media_speed   = 100_Half_Duplex

 # define the machine "test1"
 # Take all defaults and include a comment.
 test1:
     comments      = "This machine is a test machine."
# define a machine with a VIPA interface that uses interfaces en2 and en3.
 lab2:
     machine_type         = secondary
     interface_type       = vi
     interface_name       = vi0
     netaddr              = 9.53.153.235
     subnet_mask          = 255.255.255.0
     secondary_hostname   = lab3
     interface_attributes = "interface_names=en2,en3"

# define a machine with an etherchannel adapter that uses the adapters at
# the following location codes P1-I4/E1 and P1/E1
 lab4:
     machine_type         = etherchannel
     interface_type       = en
     interface_name       = en2
     netaddr              = 9.53.153.237
     subnet_mask          = 255.255.255.0
     multiple_physloc     = P1-I4/E1,P1/E1

# define a machine with an etherchannel adapter that uses the
# ent2 and ent3 adapters and uses mode 8023ad.
 lab6:
     machine_type         = etherchannel
     interface_type       = en
     interface_name       = en2
     netaddr              = 9.53.153.239
     subnet_mask          = 255.255.255.0
     adapter_attributes = "adapter_names=ent2,ent3 mode=8023ad"
```

Using secondary adapter file keywords:

The secondary adapter file uses these keywords to specify machine attributes.

Using required adapter attributes:

The following attributes are required for configuring adapters.

machine_type = secondary | etherchannel | install
Specifying the **machine_type** attribute as **secondary** clearly distinguishes the **nimadapters** input from **nimdef** input. If a secondary adapter's file is mistakenly passed to the **nimdef** command, the error can be detected. Stanzas with a **machine_type** of **install** are ignored.

netaddr
Specifies the network address for the secondary adapter.

interface_type = en | et | sn | ml |vi
Specifies the type of network interface. The network interface can be **en** (ethernet interface), **et** (ethernet interface), **sn** (switch network interface), **ml** (multi-link interface), or **vi** (virtual interface). This attribute replaces the deprecated **network_type** attribute.

subnet_mask
Specifies the subnet mask used by the secondary adapter.

Note: Configuring a secondary adapter on the same subnet as another adapter does not provide failover. Packets alternate between adapters when they are configured on the same subnet. If one of the adapters fails, the other adapter will not take over the failed adapter's workload, and the subnet will have connectivity problems. Commands, such as **mount**, might fail if this occurs.

Using optional attributes:

The following attributes are optional for configuring adapters.

adapter_attributes
Blank-separated list of physical adapter attributes and values. For example, *Attribute1=Value1 Attribute2=Value2*. To see the list of attributes that can be set for the requested adapter, run the command **lsattr -E -l** *AdapterName*.

interface_attributes
Blank-separated list of interface attributes and values. For example, *Attribute1=Value1 Attribute2=Value2*. To see the list of attributes that can be set for the requested interface, run the command **lsattr -E -l** *InterfaceName*. This attribute replaces the deprecated **attributes** attribute.

cable_type
Specifies the cable type (optional if **network_type** is **en** or **et**).

comments
Specifies a comment to include in the secondary adapter definition. Enclose the comment string in quotation marks.

interface_name
Specifies the name of the network interface for the secondary adapter (for example, **en1**, **sn0**, **ml0**). Do not specify both **location** and **interface_name**.

Note: The value of the **interface_name** attribute must be consistent with the value of the **network_type** attribute.

location
Specifies the physical location of the adapter corresponding to this network interface. Do not specify both the **location** and the **interface_name** attributes.

Note: Except for the multilink pseudo-device, use of the **location** attribute is highly recommended. If the **location** attribute is not specified and the user adds multiple adapters or adds an adapter at the same time that the operating system is reinstalled, the adapter and network interface names might be reassigned by the operating system in unexpected ways.

multiple_physloc
Specifies the physical adapters to associate with an interface when you use an etherchannel or VIPA stanza.

media_speed
Specifies the media speed (optional if the **network_type** attribute's value is either **en** or **et**).

secondary_hostname
Host name to save in the **/etc/hosts** file with the **netaddr** attribute. This host name is not set using the **hostname** command or the **uname -S** command.

Working with secondary adapter file rules:

The format of the secondary adapter file must comply with these rules.

- After the stanza header, follow attribute lines of the form: *Attribute = Value*
- If you define the value of an attribute multiple times within the same stanza, only the last definition is used.
- If you use an invalid attribute keyword, that attribute definition is ignored.
- Each line of the file can have only one header or attribute definition.
- More than one stanza can exist in a definition file for each machine host name.
- Each stanza for a machine host name represents a secondary adapter definition on that NIM client. No two secondary adapter definitions for the same machine host name can have the same location or **interface_name**. There should be only one definition per adapter or interface on a given NIM client.
- If the stanza header entry is the **default** keyword, this specifies to use that stanza for the purpose of defining default values.
- You can specify a default value for any secondary adapter attribute. However, the **netaddr** and **secondary_hostname** attributes must be unique. Also, the **location** and **interface_name** attributes must be unique on a NIM client.
- If you do not specify an attribute for a secondary adapter but define a default value, the default value is used.
- You can specify and change default values at any location in the definition file. After a default value is set, it applies to all definitions that follow.
- To turn off a default value for all following machine definitions, do not set the attribute value in a default stanza.
- To turn off a default value for a single machine definition, do not set the attribute value in the machine stanza.
- You can include comments in a client definition file. Comments begin with the number sign (**#**).
- When parsing the definition file for header and attribute keywords and values, tab characters and spaces are ignored.

Note: During a **nim -o bos_inst** or **nim -o cust** operation, if NIM examines the configuration data on the client and determines that a secondary adapter is already configured with precisely the attributes requested in the **adapter_def** resource, this secondary adapter is not reconfigured.

Working with secondary adapter definitions:

Follow these procedures to work with NIM secondary adapter definitions.

1. To preview the **secondary_adapters.defs** client definition file, type:
   ```
   nimadapters -p -f secondary_adapters.defs adapter_def
   ```
2. To add the NIM secondary adapters described in the **secondary_adapters.defs** secondary adapters definition file, type:
   ```
   nimadapters -d -f secondary_adapters.defs adapter_def
   ```
3. To define the NIM secondary adapters for the **pilsner** client, type:

```
nimadapters -d \
-a info="en,P2-I1/E1,N/A,1000_Full_Duplex,9.53.153.233,255.255.254.0" \
-a client=pilsner adapter_def
```

4. To remove the NIM secondary adapter definitions for a client called pilsner from the **my_adapter_def** resource, type:

   ```
   nimadapters -r -a client=pilsner my_adapter_def
   ```

5. To remove the NIM secondary adapter definitions for clients defined in the file **secondary_adapters.defs**, type:

   ```
   nimadapters -r -f secondary_adapters.defs my_adapter_def
   ```

6. To remove all the NIM secondary adapter definitions from the **my_adapter_def** resource, type:

   ```
   nimadapters -r my_adapter_def
   ```

Troubleshooting secondary adapter file stanza errors:

A secondary adapter stanza causes an error under any of the following conditions.

- The host name that was used in the stanza header for the definition cannot be resolved.
- A required attribute is missing.
- An invalid value was specified for an attribute.
- An attribute mismatch occurs. For example, if the **network_type** attribute's value is not set to either **en** or **et**, you cannot specify **cable_type=bnc** or **media_speed=1000_Full_Duplex**.
- The stanza contains both a **location** attribute and an **interface_name** attribute.
- Secondary adapter definitions occur multiple times for the same adapter location and the same host name.
- Secondary adapter definitions occur multiple times for the same **interface_name** and the same host name.

If a secondary adapter stanza is incorrect, the errors are reported, the stanza is ignored, and the following input is processed without regard to the incorrect stanza.

Using the NIM service handler for client communication

NIM makes use of the remote shell server (rshd) when it performs remote execution on clients. The server provides remote execution facilities with authentication based on privileged port numbers from trusted hosts.

AIX uses NIM Service Handler (NIMSH) to eliminate the need for rsh services during NIM client communication. The NIM client daemon (NIMSH) uses reserved ports 3901 and 3902, and it installs as part of the **bos.sysmgt.nim.client** fileset.

NIMSH allows you to query network machines by hostname. NIMSH processes query requests and returns NIM client configuration parameters used for defining hosts within a NIM environment. Using NIMSH, you can define NIM clients without knowing any system or network-specific information.

While NIMSH eliminates the need for rsh, it does not provide trusted authentication based on key encryption. To use cryptographic authentication with NIMSH, you can configure OpenSSL in the NIM environment. When you install OpenSSL on a NIM clients, SSL socket connections are established during NIMSH service authentication. Enabling OpenSSL provides SSL key generation and includes all cipher suites supported in SSL version 3.

Using NIMSH:

Basic NIMSH functions are explained.

NIMSH service port:

The client daemon has two ports registered with the Internet Assigned Numbers Authority (IANA) for use during network communication. These ports are referred to as the primary and secondary ports.

The **nimsh** client daemon listens on these ports for requests initiated by the master using the TCP protocol. The primary port listens for service requests on reserved port 3901. When a request is accepted, the primary port is used for **stdin** and **stdout** requests. The **stderr** requests are redirected to secondary port 3902. This behavior is similar to auxiliary connections in **rcmd()**. This implementation allows the NIM master connection to stay consistent with current support of client connections through the **rsh** command. Using a reserved secondary port in NIMSH allows firewall administrators to write firewall rules for accepting incoming connections on privileged ports from the secondary port. These rules can have the requirement that the originating socket address (hostname : secondary port) comes from a trusted source.

NIMSH system resource control:

NIMSH is registered with the System Resource Controller (SRC). The SRC group name is **nimclient** and the subsystem defined is NIMSH.

The client daemon is started by SRC when the configuration routine is run using the **nimclient** command.

NIMSH authentication process:

Service requests from the communicating host (the NIM master) will build packets with the following data for authentication.

- Hostname of NIM client
- CPUID of NIM client
- CPUID of NIM master
- Return port for secondary (**stderr**) connection
- Query flag (used to obtain registration information)

When a connection to the primary port is received, the service handler obtains peer information from the connecting socket. The source port must be in the privileged port space (only root user can bind to privileged ports). Using the privileged port space ensures that the originating user has the root UID. The return port number is retrieved and connected to from the secondary port, which is the reserved port in **/etc/services**.

The following sections describe the phases of the authentication process:

query flag set

When the **query** flag is set to 1, the service handler treats the incoming request as a client discovery for information. The client service handler obtains all relevant information necessary for defining itself as a NIM client and returns the information to the requesting NIM master then terminates the connection. The following data is returned when query flag is set:

- Default hostname (value obtained from **inet0**)
- Default route (value obtained from **inet0**)
- Network address (value obtained from hostname)
- Subnet mask (value obtained from hostname)
- Network interface (value obtained from hostname)

Method request

If the query flag is not set, then a request for service (NIM operation) is pushed by the NIM master. The service handler validates the method request as follows:

1. Verify hostname of NIM master is the client's recognized master hostname.
2. Check the client CPUID passed, it should match the client's machine ID.
3. Check the master CPUID passed, it should match the master's machine id stored in memory.
4. Verify the operation passed is a method in the path **/usr/lpp/bos.sysmgt/nim/methods**.
5. Check for cryptographic authentication setting.

For additional security, NIMSH supports push disablement. Push disablement disables method requests unless the request is a response to an active NIM client call being processed by the client's NIM master. When push disablement is set, NIMSH does not process any NIM operations controlled by the NIM master. Client control (**nimclient** commands) is the only way to perform NIM operations on the client when push disablement is set.

Logging NIMSH operations:

The NIM client daemon logs data in the **/var/adm/ras/nimsh.log** file during its operation. The log is used only for debug purposes.

Setting up NIMSH:

You can configure existing standalone clients to use NIMSH as the communication protocol. You can also define clients using NIMSH as the service option.

For more information on doing this, see Adding a Standalone NIM client to the NIM Environment.

Preparing to set up NIMSH:

These prerequisites must be met prior to configuring NIMSH.

- The NIM client must already be configured (see "Adding standalone clients to the NIM environment" on page 134).
- The client and the client's NIM master must have one of the following installed:
  - AIX 5.2 with the 5200–07 Technology Level (or later)
  - AIX 5.3 with the 5300–03 Technology Level (or later)
  - AIX 6.1 or later

Setting up NIMSH using SMIT:

Follow this procedure for configuring existing standalone clients with NIMSH using SMIT.

Complete the following steps:

1. Type the `smitty nim_config_services` fast path on the NIM client.
2. Select **nimsh** as the Communication Protocol used by client.

Setting up NIMSH from the command line:

Rename the /etc/niminfo file to another name on the NIM client. You also can delete it, but renaming it allows you to keep a copy if you should need it for some reason.

Use the **niminit** command to register the client with the NIM master.

```
# niminit -a name=<client_name> -a master=<master_name> -a connect=nimsh
```

If OpenSSL is installed on the NIM client and NIMSH is configured as the communication protocol, type the following command on the NIM client to disable cryptographic authentication with NIMSH.

Type the following command on the NIM client:

```
# nimclient -C
```

Enabling cryptographic authentication:

You can configure existing standalone clients to use the NIMSH communication protocol with SSL enabled.

NIM supports OpenSSL versions 0.9.6e and higher. When OpenSSL is installed, NIMSH uses SSL-encrypted certificates for authenticating the connecting NIM master.

Preparing to enable cryptographic authentication:

These prerequisites must be met to enable cryptographic authentication.

- The NIM master must already be configured for SSL authenticating within the NIM Environment. For more information, see "Using NIM to install clients configured with SSL authentication" on page 180.
- The client must be at AIX 5.3 or later version.
- The client's NIM master must be at AIX 5.3 or later version.

Enabling cryptographic authentication using SMIT:

Complete these steps to configure existing standalone clients to use NIMSH communication protocol with SSL enabled.

1. Type the **smitty nim_config_services** fast path on the NIM client.
2. Select **nimsh** as the Communication Protocol used by client.
3. Select **enabled** as the option for **Enabling Cryptographic Authentication**.
4. Select **yes** as the option for **Installing Secure Socket Layer Software**, if OpenSSL is not installed on the client.
5. Specify the absolute path for the RPM package or select the **lpp_source** resource that contains the OpenSSL RPM package.

Enabling cryptographic authentication from the command line:

Complete these steps to configure existing standalone clients to use the NIMSH communication protocol with SSL enabled from the command line.

- If OpenSSL is installed on the NIM client and NIMSH is configured as the communication protocol, type the following command:

  ```
  # nimclient -c
  ```

- If OpenSSL is not installed on the NIM client, complete the following steps
  1. Install OpenSSL if not already installed. It can be installed from the base media using the installp command, geninstall command, or smitty command.
  2. Type the following command on the NIM client after OpenSSL is installed:

     ```
     # nimclient -c
     ```

- If OpenSSL is installed on the NIM alternate master, type the following command to configure NIMSH as the communication protocol:

  ```
  # nimclient -c
  # nimconfig -c
  ```

- If the NIM client running NIMSH with OpenSSL communication protocol wishes to communicate with an alternate master running NIMSH with OpenSSL encryption, type the following command on the NIM client where <alternate_master> is the NIM object name of the alternate_master:

  ```
  # nimclient -o get_cert -a master_name=<alternate_master>
  ```

Enabling a secondary port:

This procedure describes how to configure existing standalone clients to use the NIMSH communication protocol with a secondary port option enabled.

By default, NIMSH uses a reserved port for returning **stderr** output during command execution. The default setting allows administrators to specify a specific port for opening behind a firewall, but it can cause performance issues when several connections are attempted in a short amount of time.

When TCP connections are closed, the closing sockets enter **TIME_WAIT** state. The length of time for this state may last up to 240 seconds depending on system settings. The secondary port option allows you to specify any specific range of ports to cycle through during NIMSH operation.

For firewalls, administrators might want to open a specific range on the firewall, and then for each machine on the internal network, ensure that the port range on the machine coincides with the open range on the firewall. When changing the NIMSH secondary port, you should choose a range of ports outside of the range used for system services. Try using ports 49152 through 65535.

Preparing to enable a secondary port:

These prerequisites must be met to enable a secondary port.

- The NIM client must already be configured (see Adding a Standalone NIM client to the NIM Environment).
- The client must have AIX 5.3 or later is installed.
- The client's NIM master must have AIX 5.3 or later is installed.

Enabling a secondary port from SMIT:

Complete these steps to configure existing standalone clients to use the NIMSH communication protocol with a secondary port range.

1. Type the **smitty nim_config_services** fast path on the NIM client.
2. Select **nimsh** as the Communication Protocol used by client.
3. Specify a start value for the secondary port number.
4. Specify an increment value for the secondary port range.

Enabling a secondary port from the command line:

Complete these steps to configure existing standalone clients to use the NIMSH communication protocol with a secondary port range from the command line.

1. Edit the **/etc/environment** file.
2. Add the variable *NIM_SECONDARY_PORT=60000:5*, to use ports 60000 - 60005 within NIMSH.
3. Use the desired **nimclient** command option to restart the NIMSH daemon.

Disabling push operations using NIMSH:

NIM clients can prohibit the NIM master from allocating resources or initiating operations by disabling push operations.

Although master control is disabled, the client can still control the allocation of NIM resources and the initiation of NIM operations. To configure existing standalone clients to use NIMSH communication protocol with NIM master control disabled, see “Disabling master push permissions in the NIM environment” on page 209.

Verifying NIMSH startup:

Run this command to verify that the NIMSH daemon is enabled on the client.

```
# lssrc -s nimsh
```

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 SMIT, or the **nim -o change** command operation.

Creating additional interface attributes from SMIT:

Follow this procedure for creating additional interface attributes 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.

Creating additional interface attributes from the command line:

Use this procedure for creating additional interface attributes 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 AdapterHostNam
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 108.

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
```

Creating network boot images to support only the defined clients and networks

You can create network boot images in the **/tftpboot** directory using the SMIT, or the command line interface.

When a SPOT resource is created, network boot images are created in the `/tftpboot` directory to support certain NIM operations.

NIM only creates 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.

Creating network boot images to support defined clients and networks using SMIT:

Use this method to manage the creation of boot images from the SMIT interface.

Type the SMIT fast path:

```
smit nim_control_boot
```

Creating network boot images to support defined clients and networks from the command line:

Use this information to manage network boot images to support only the defined clients and networks.

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
```

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.

Using client machines as resource servers using SMIT:

Follow this procedure for using client machines as resource servers using 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.

Using client machines as resource servers from the command line:

Follow this procedure for using client machines as resource servers 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
```

Using concurrency control

Users can ease the severity of NIM installations becoming overburdened when they are being performed on a large number of clients at the same time by controlling the number of clients that are installed.

NIM installations can become overburdened 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.

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. After 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.

Using concurrency control from SMIT:

You can access concurrency control attributes from all SMIT panels under the Install and Update Software menu and the Alternate Disk Installation menu.

Using concurrency control 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 **bos.games** fileset installed on only five 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 three 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 the following states:

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

In the first situation, the group will revert 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 rerunning the command on the group. Alternatively, the user can "reset" the group, which will set the group back to its state prior to the concurrency controlled operation.

Migrating the Virtual I/O Server using NIM

You can use the following procedures to perform a migration installation of the Virtual I/O Server into environments managed by the HMC or Integrated Virtualization Manager using NIM.

Prerequisites

The Virtual I/O Server installation media is required.

In addition, the following system requirements must be met:

- A system running AIX 5.3 with 5300-03 or higher which contains a file system with at least 700 MB available.

- A logical partition of type Virtual I/O Server containing an Ethernet adapter connected to an active network for installing the Virtual I/O Server. For information about creating logical partitions, see Creating the Virtual I/O Server logical partition and partition profile.
- A storage controller containing at least 16 GB of disk space.

After the prerequisites have been met, follow these steps to use NIM to migrate the Virtual I/O Server:

1. Insert the *Virtual I/O Server Migration* DVD into the DVD drive.
2. Run the **installios** command without any arguments to start the installation wizard. The **installios** wizard then guides you through the process of filling-out the necessary information to start an installation on the Virtual I/O Server or on an Integrated Virtualization Manager.

   If you run **installios** on a NIM client, then you are prompted for the location to the **bos.sysmgt.nim.master** fileset. The NIM client is then configured as a NIM master. For more information about command-line usage of **installios**, see the **installios** command.

   The **installios** setup process creates the following NIM resources to start the migration installation:
   - bosinst_data
   - installp_bundle
   - lpp_source
   - resolv_conf
   - SPOT
   - Client definition

   If you are installing the Virtual I/O Server logical partition, and if Secure Shell (SSH) and credentials have been configured on the NIM master, then the partition is network-booted from the HMC to begin the installation.

   If you are installing the Virtual I/O Server logical partition without SSH, or if you are installing the Integrated Virtualization Manager, then go to step 3.
3. On the system on which the Virtual I/O Server software will be installed, boot the Virtual I/O Server logical partition or the Integrated Virtualization Manager into System Management Services (SMS) mode by following these steps:
   - To boot the Virtual I/O Server logical partition into SMS:
     a. On the HMC, right-click the partition to open the menu.
     b. Click **Activate**. The Activate Partition menu opens with a selection of partition profiles. Be sure the correct profile is highlighted.
     c. Select the **Open a terminal window or console session** check box to open a virtual terminal (vterm) window.
     d. Click **(Advanced...)** to open the advanced options menu.
     e. For the Boot mode, select **SMS**.
     f. Click **OK** to close the advanced options menu.
     g. Click **OK**. A vterm window opens for the partition.
     h. In the vterm window, select **Setup Remote IPL** (Initial Program Load).
     i. Select the network adapter that will be used for the installation.
     j. Select **IP Parameters**.
     k. Enter the client IP address, server IP address, and gateway IP address. Optionally, you can enter the subnet mask. After you have entered these values, press Esc to return to the Network Parameters menu.
     l. Select **Ping Test** to ensure that the network parameters are properly configured. Press Esc twice to return to the Main Menu.
     m. From the Main Menu, select **Select Boot Options**.
     n. Select **Select Install/Boot Device**.

o. Select **Network**.
p. Select the network adapter whose remote IPL settings you previously configured.
q. When prompted for **Normal** or **Service** mode, select **Normal**.
r. When asked if you want to exit, select **Yes**.

- To boot the Integrated Virtualization Manager into SMS:
  a. Begin with the machine turned off.
  b. Switch on the machine, and as icons begin to appear from left to right on the bottom of your display, press F1.

     Note: If the last icon is displayed before pressing F1, then you get the normal mode boot list instead of SMS, so try again.
  c. The System Management Services menu opens. Select **Utilities**.
  d. From the System Management Services Utilities menu, select **Remote Initial Program Load Setup**.
  e. From the Network Parameters panel, select **IP Parameters**.
  f. Set or change the displayed values so they are correct for your client system. Specify the IP address of the following:
     – The client machine you are booting in the client address field.
     – Your NIM master server in the server address field.
     – Your client's gateway in the gateway address field.
     – Your client's subnet mask in the subnet mask field.
  g. After you specify the addresses, press Enter to save the addresses and continue.
  h. The Network Parameters window opens. Select the Ping option.
  i. Select the network adapter to be used as the client's boot device.
  j. Verify that the displayed addresses are the same as the addresses you specified for your boot device. If the addresses are incorrect, press Esc until you return to the main menu. Then, go back to step e. If they are correct, continue with k.
  k. If the addresses are correct, press Enter to perform the ping test. The ping test might take several seconds to complete.
  l. If the ping test fails, verify that the addresses are correct, and analyze the network problem. If the ping test is successfull, press Enter to acknowledge the success message.
  m. Press Esc until you return to the System Management Services menu.
  n. From the System Management Services menu, choose the **Select Boot Devices** option.
  o. Select the network adapter to be used for the network boot from the list of displayed bootable devices

After the migration installation is complete, the Virtual I/O Server logical partition or the Integrated Virtualization Manager is booted to its configuration prior to the migration installation.

To remove all the NIM resources that were created from the **installios** setup process, run the **installios** command with the **-u** flag. If the **installios** command fails to perform the cleanup, run **installios -u** and specify the **-f** flag to force NIM to reset and deallocate resources to the client. The NIM environment still exists, but all resources and directory structures created from the **installios** wizard are removed. If, however, you want to unconfigure NIM, or uninstall the **bos.sysmgt.nim.master** fileset and return the NIM master to a NIM client (if it was configured from a NIM client), specify **installios -u** with a **-U** flag.

Installing with NIM

You can use Network Installation Management (NIM) to manage the installation of the Base Operating System (BOS) for multiple configurations and locations.

Using installation images to install the base operating system on a NIM client

Using installation images to install the base operating system (BOS) on a NIM client is similar to the traditional BOS installation from a tape or media 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 121.
- The NIM client to be installed must already exist in the NIM environment. To add the client to the NIM environment, see "Adding standalone clients to the NIM environment" on page 134.

Using installation images to install the base operating system on a NIM client using SMIT:

Follow this procedure to install use installation images to install the base operating system a NIM client using SMIT.

1. To install BOS on a NIM client using an **rte** installation, type `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. If you are booting from a network device, follow the procedures in your hardware documentation to perform 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 nonprompted installation, follow the instructions in "Performing a nonprompted BOS installation" on page 168 to complete the prerequisite tasks.

Using installation images to install the base operating system on a NIM client from the command line:

Follow this procedure for using installation images to install the base operating system on a NIM client from the command line.

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

   ```
   # 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. If you are booting from a network device, follow the procedures in your hardware documentation 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
   ```

   Note:

   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 nonprompted installation, follow the instructions in "Performing a nonprompted BOS installation" to complete the prerequisite tasks.
   c. For a complete description of the different ways that a BOS installation can be customized by NIM, see "Using the NIM bos_inst operation" on page 262.

Performing a nonprompted BOS installation

This procedure provides information about how to create a **bosinst_data** resource to use for a nonprompted BOS installation.

After you have created the **bosinst_data** resource, refer to the following procedures to perform the nonprompted installation:

- "Using installation images to install the base operating system on a NIM client" on page 167
- "Using a mksysb image to install the base operating system on a NIM client" on page 169

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 121.
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 standalone clients to the NIM environment" on page 134.
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. For a sample `bosinst.data` file that specifies the syntax to control license acceptance, see "Using the bosinst.data file" on page 56.

Performing a nonprompted BOS installation using SMIT:

You can perform a nonprompted BOS installation using 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. For a sample **bosinst.data** file, see "Using the bosinst.data file" on page 56.
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** or **mksysb** installation on a standalone machine. Be sure to specify the **bosinst_data** resource to use during the installation.

Performing a nonprompted BOS installation from the command line:

You can perform a nonprompted BOS 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.

   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** or **mksysb** installation on standalone machines. Be sure to specify that the **bosinst_data** resource be used for the installation.

   For example, to perform a nonprompted **rte** installation 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
   ```

Using a mksysb image to install the base operating system on a NIM client

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

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).

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

Note: A NIM customization that affects the ODM database is not reflected after a **mksysb** installation on the same system. Restoring the **mksysb** backup causes the ODM to be restored to the state it was in when the backup was created.

In AIX, devices are not recovered if the **mksysb** image that is being installed was not created on the same system.

Attention: If the system that you cloned is using OpenGL, there might be device filesets that must be installed after a clone. OpenGL has graphics adapter-specific filesets, so if you cloned onto a system with a different graphics adapter, you need to create a bundle as follows:

```
echo OpenGL.OpenGL_X.dev > /usr/sys/inst.data/user_bundles/graphic_dev.bnd
```

You can allocate this bundle when you install the **mksysb**, and the device filesets are installed automatically if OpenGL is in your **lpp_source**.

Prerequisites

- The NIM master must be configured, and **SPOT** and **mksysb** resources must be defined.
- The NIM client to be installed must already exist in the NIM environment.
- The **mksysb** must be available on the hard disk of the NIM master or a running NIM client, or the **mksysb** image is created during this procedure from either the NIM master or a running NIM client.
- The **SPOT** and **mksysb** resources should be at the same level of AIX when used for NIM BOS installations.
- 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 situation 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.

Related concepts:

"Using the NIM bos_inst operation" on page 262
Use the **bos_inst** operation to install the AIX BOS on standalone clients.

"Configuring the NIM master and creating basic installation resources" on page 121
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 with SMIT, or the command line.

"Adding standalone clients to the NIM environment" on page 134
You can add standalone clients to the NIM environment with SMIT, or the command line.

Using a mksysb image to install the base operating system on a NIM client using SMIT:

Follow this procedure for using a mksysb image to install the base operating system on a NIM client using 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. 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.
12. Run the SMIT dialog to install the NIM client.

13. 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. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.
14. 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 nonprompted installation, follow the instructions in "Performing a nonprompted BOS installation" on page 168 to complete the prerequisite tasks.

Using a mksysb image to install the base operating system on a NIM client from the command line:

Follow this procedure for using a mksysb image to install the base operating system on a NIM client 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 "Using a mksysb resource" on page 245.

   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 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** 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. If you are booting from a network device, follow the procedures in your hardware documentation 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`, an optional **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 nonprompted installation, follow the instructions in "Performing a nonprompted BOS installation" on page 168 to complete the prerequisite tasks.

Using an ios_mksysb image to install the base operating system on a NIM client from the command line:

Procedure for using an **ios_mksysb** image to install the base operating system on a NIM client from the command line.

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

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

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

   For a complete description of all the options that can be specified when creating a **ios_mksysb** resource, see "Using an ios_mksysb resource" on page 240.

   Example 1:

   To define a **ios_mksysb** resource, `ios_mksysb_res1`, from an existing **ios_mksysb** image that is located in the **/export/backups/client_ios_mksysb** on the master, enter:

   ```
   nim -o define -t ios_mksysb -a server=master \
   -a location=/export/backups/client_ios_mksysb ios_mksysb_res1
   ```

   Example 2:

   To create a **ios_mksysb** image of the client system, `client1`, in `/export/resources/new_ios_mksysb` on the master, and to define an **ios_mksysb** resource, `ios_mksysb_res2`, enter:

   ```
   nim -o define -t ios_mksysb -a server=master \
   -a location=export/resources/new_ios_mksysb -a mk_image=yes \
   -a source=client1 ios_mksysb_res2
   ```

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

   ```
   nim -o bos_inst -a source=mksysb -a ios_mksysb=ios_mksysb \
   -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 **ios_mksysb** and **SPOT** resources.

   If the client system being installed is not already a running, configured NIM client, NIM will not automatically reboot the system over the network for installation. A network boot must be performed manually on the system. 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 VIOS or IVM client was not rebooted automatically, initiate a network boot from the client to install it. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.

4. After the system boots over the network, the display on the client system will begin prompting for information about how to configure the system during installation. Specify the requested information to continue with the installation.
5. If the client was not rebooted automatically, initiate a network boot from the client to install it. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.
6. After the system boots over the network, the display on the client system will begin prompting for information about how to configure the system during installation. Specify the requested information to continue with the installation.

   Example 3:

   To perform a **mksysb** installation using the **ios_mksysb1**, `mksysb1`, an optional **lpp_source**, `lpp_source1`, and the **SPOT**, `spot1`, on client system, `machine1`, which is not a running, configured NIM client, enter:

   ```
   nim -o bos_inst -a source=mksysb -a ios_mksysb=ios_mksysb1 \
   -a lpp_source=lpp_source1 -a spot=spot1 -a boot_client=no vios1
   ```

Notes:

1. The steps to perform an **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 an **ios_mksysb** resource must be allocated for the operation.
2. To perform a non-prompted installation, follow the instructions in "Performing a nonprompted BOS installation" on page 168 to complete the prerequisite tasks.

Using an ios_mksysb image to install the base operating system on a NIM client using SMIT:

Follow this procedure for using an **ios_mksysb** image to install the base operating system on a NIM VIOS management client using SMIT.

1. If the **ios_mksysb** resource has already been created, skip to step 6. Otherwise, to create the **ios_mksysb** resource, enter the **smit nim_mkres** fast path.
2. Select **ios_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 **ios_mksysb** resource.
4. If the **ios_mksysb** image does not exist, create it by supplying the values for the fields under **System Backup Image Creation Options**.

   Note: If the **ios_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 **ios_mksysb** resource.
5. Upon successful completion of this task, exit SMIT.
6. To use the **ios_mksysb** resource to install a NIM client, enter the **smit nim_mgmt_obj_op** fast path.
7. Select a TARGET for the operation.
8. Select **bos_inst** for the operation to perform.
9. In the displayed dialog MKSYSB field, select the **ios_mksysb** resource.
10. In the displayed dialog SPOT field, select the SPOT to use for the installation. The SPOT must be one created from the **ios_mksysb** resource.
11. 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.
12. Run the SMIT dialog to install the NIM VIOS client.
13. If the VIOS client machine being installed is not already a running, configured NIM VIOS 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. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.

14. 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 nonprompted BOS installation" on page 168 to complete the prerequisite tasks.

Installing to an alternate disk on a NIM client

NIM allows you to install an AIX 4.3 or later **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 for a normal installation.

Note: For information about the different ways NIM can customize an alternate disk installation, see "Using the NIM alt_disk_install operation" on page 258.

Prerequisites

- 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 121.
- The NIM client must already exist in the NIM environment and must be running. To add the client to the NIM environment, see "Adding standalone clients to the NIM environment" on page 134.
- 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 141.

Installing to an alternate disk on a NIM client using SMIT:

Follow this procedure for installing to an alternate disk on a NIM client using 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.

Installing to an alternate disk on a NIM client from the command line:

The **alt_disk_install** command is initiated on the target system, and progress is shown with the **lsnim** command.

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:

Use this code for initiating the **alt_disk_install** operation.

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 "Using a mksysb resource" on page 245.

Cloning the rootvg to an alternate disk:

Use this command for 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.

Installing to an alternate disk on a NIM client from the command line - examples:

Here is an examples of performing an installation to an alternate disk on a NIM client from the command line.

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 `51mksysb` enter:

  ```
  nim -o alt_disk_install -a source=mksysb -a mksysb=51mksysb \
  -a disk=hdisk2 machine1
  ```

- To clone the rootvg to `hdisk2` enter:

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

Installing the Virtual I/O Server using NIM

You can use the following procedures to install the Virtual I/O Server into environments managed by the HMC or the Integrated Virtualization Manager using Network Installation Management (NIM).

Installing the Virtual I/O Server using installios:

You can use the following procedures to install the Virtual I/O Server into environments managed by the HMC or the Integrated Virtualization Manager using the **installios** command.

Prerequisites

You need the following files before beginning this procedure. These files are located on the Virtual I/O Server installation media:

- **nimol/ioserver_res/mksysb** (the mksysb image)

In addition, the following system requirements must be met:

- A file system with at least 700 MB available.
- A Virtual I/O Server logical partition that contains an Ethernet adapter connected to an active network for installing the Virtual I/O Server. For information about creating logical partitions, see Creating the Virtual I/O Server logical partition and partition profile.

- A storage controller containing at least 16 GB of disk space.

Complete the following steps to use NIM to install the Virtual I/O Server:

1. Insert the *Virtual I/O Server* DVD into the DVD drive.
2. Run the **installios** command without any arguments to start the installation wizard. The **installios** wizard guides you through the process of completing the necessary information to start an installation on the Virtual I/O Server or on the Integrated Virtualization Manager.

   If you run **installios** on a NIM client, then you are prompted for the location to the **bos.sysmgt.nim.master** fileset. The NIM client is then configured as a NIM master. For more information about command-line use of the **installios** command, see the **installios** command.

   The **installios** setup process creates the following NIM resources to start the installation:

   - bosinst_data
   - installp_bundle
   - lpp_source
   - mksysb
   - resolv_conf
   - SPOT
   - Client definition

   If you are installing the Virtual I/O Server logical partition, and if Secure Shell (SSH) and credentials have been configured on the NIM master, then the partition is network-booted from the Hardware Management Console (HMC) to begin the installation.

   If you are installing the Virtual I/O Server logical partition without SSH, or if you are installing the Integrated Virtualization Manager, then go to step 3.
3. On the system on which the Virtual I/O Server software will be installed, boot either the Virtual I/O Server logical partition or the Integrated Virtualization Manager into System Management Services (SMS) mode by following these steps:
   - To boot the Virtual I/O Server logical partition into SMS:
     a. On the HMC, right-click the partition to open the menu.
     b. Click **Activate**. The Activate Partition menu opens with a selection of partition profiles. Be sure the correct profile is highlighted.
     c. Select the **Open a terminal window or console session** check box to open a virtual terminal (vterm) window.
     d. Click **(Advanced...)** to open the advanced options menu.
     e. For the Boot mode, select **SMS**.
     f. Click **OK** to close the advanced options menu.
     g. Click **OK**. A vterm window opens for the partition.
     h. In the vterm window, select **Setup Remote IPL** (Initial Program Load).
     i. Select the network adapter that will be used for the installation.
     j. Select **IP Parameters**.
     k. Enter the client IP address, server IP address, and gateway IP address. Optionally, you can enter the subnet mask. After you have entered these values, press Esc to return to the Network Parameters menu.
     l. Select **Ping Test** to ensure that the network parameters are properly configured. Press Esc twice to return to the Main Menu.
     m. From the Main Menu, select **Select Boot Options**.
     n. Select **Select Install/Boot Device**.
     o. Select **Network**.
     p. Select the network adapter whose remote IPL settings you previously configured.

q. When prompted for **Normal** or **Service** mode, select **Normal**.
r. When asked if you want to exit, select **Yes**.

- To boot the Integrated Virtualization Manager into SMS:
  a. Begin with the machine turned off.
  b. Turn on the machine, and as icons begin to appear from left to right on the bottom of your display, press F1.

     Note: If the last icon is displayed before pressing F1, you get the normal mode boot list instead of SMS. Repeat steps a and b.
  c. The System Management Services menu opens. Select **Utilities**.
  d. From the System Management Services Utilities menu, select **Remote Initial Program Load Setup**.
  e. From the Network Parameters panel, select **IP Parameters**.
  f. Set or change the displayed values so they are correct for your client system. Specify the IP address of the following, then press Enter:
     – The client machine you are booting in the client address field.
     – Your NIM master server in the server address field.
     – Your client's gateway in the gateway address field.
     – Your client's subnet mask in the subnet mask field.
  g. When the Network Parameters window opens, select the Ping option.
  h. Select the network adapter to be used as the client's boot device.
  i. Verify that the displayed addresses are the same as the addresses you specified for your boot device. If the addresses are incorrect, press Esc until you return to the main menu. Then, go back to Step e. If they are correct, continue with Step j.
  j. Press Enter to perform the ping test. The ping test might take several seconds to complete.
  k. If the ping test fails, verify that the addresses are correct, and analyze the network problem. If the ping test is successful, press Enter to acknowledge the success message.
  l. Press Esc until you return to the System Management Services menu.
  m. From the System Management Services menu, choose the **Select Boot Devices** option.
  n. Select the network adapter to be used for the network boot from the list of displayed bootable devices.

After the installation is complete, the Virtual I/O Server logical partition or the Integrated Virtualization Manager is ready to be configured and managed.

To remove all of the NIM resources that were created from the **installios** setup process, run the **installios** command with the **-u** flag. If the **installios** command fails to perform the cleanup, run **installios -u** and specify the **-f** flag to force NIM to reset and deallocate resources to the client. The NIM environment remains, but all of the resources and directory structures created from the **installios** wizard are removed. If, however, you want to unconfigure NIM, or to uninstall the **bos.sysmgt.nim.master** fileset and return the NIM master back to a NIM client if it was configured from a NIM client, specify **installios -u** with a **-U** flag.

You can also install a Virtual I/O Server or an Integrated Virtualization Manager through the SMIT interface.

1. To access the SMIT interface to the **installios** command, run `smitty installios` on a NIM master.
2. You will have two options: to **Setup for Virtual I/O and Integrated Virtualization Manager Installation** and **Cleanup after Virtual I/O and Integrated Virtualization Manager Installation**, where **Configure Client as Master for Virtual I/O and Integrated Virtualization Manager Installation** is the only available option on a NIM client.

3. Complete the required fields from the **installios** wizard to invoke the **installios** command that will setup the installation or perform a cleanup.

Installing the Virtual I/O Server using ios_mksysb:

You can use the following procedures to install the Virtual I/O Server (VIOS) into environments managed by the Hardware Management Console (HMC) or the Integrated Virtualization Manager using the **ios_mksysb** resource.

Prerequisites

- The Network Installation Management (NIM) master must be configured, and SPOT and **mksysb** resources must be defined. See "Configuring the NIM master and creating basic installation resources" on page 121.

  The **mksysb** resource can be created from a VIOS NIM management client.

  Note: The **mksysb** on the VIOS media is split into multiple files due to the file size constraint when you generate the **mksysb** file for the VIOS media. The split **mksysb** files need to be joined together when copied from media to the hard disk of the system. The following example shows the VIOS media mounted to the **/mnt** directory.

  ```
  cat /mnt/nimol/ioserver_res/mksysb \
  /mnt/nimol/ioserver_res/mksysb2 > /export/mksysb/vio_mksysb
  ```

  The **mksyb** file may be split across multiple VIOS media. In that case, the **mksysb** file must be concatenated into a single file using the **cat** command describe in the example above from a multivolume VIOS media.

  Define the **mksysb** file as a NIM **ios_mksysb** resource.

  The **mksysb** image can also be created from the VIOS by using the following command:

  ```
  nim -o define -t ios_mksysb
  ```

  See "Defining the mksysb resource" on page 245.
- The NIM VIOS client to be installed must already exist in the NIM environment. To add the client to the NIM environment, see "Adding VIOS management objects to the NIM environment" on page 130.
- The SPOT resource must be created from the **ios_mksysb** resource. To do this, define the SPOT resource, by specifying a **ios_mksysb** NIM object as the value for the source attribute.
- The **bosinst_data** resource can be copied from the VIOS media and defined as a NIM **bosinst_data** resource or defined as new. To define a new **bosinst_data** resource, copy the **bosinst.data** template from a system at **/usr/lpp/bos.inst/bosinst.template** and set **RECOVER_DEVICES=Default**. If the **ios_mksysb** resource is to be deployed to a specific disk then the **target_disk_data** section of the **bosinst.data** must be populated with disk information from the VIOS server. For more information about the **bosinst.data**, consult the documentation about the **bosinst.data** file.
- An **ios_mksysb** installation restores the VIOS and the Base Operation System to a Virtual I/O Server.
- The **ios_mksysb** images enable you to clone one system image onto multiple target systems.

Using NIM to install clients configured with Kerberos authentication

You can install clients configured with Kerberos authentication using NIM.

Normally, NIM relies on Standard AIX authentication to allow the NIM master to remotely execute commands. Standard AIX authentication uses 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 4.3.2 and later, NIM can be used to install machines in an 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 3.1 (or later) 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. Because NIM will not have configured Kerberos 4 on the client after the BOS is installed. NIM will therefore have to rely on a `.rhosts` file 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, 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.

For more information on installing and configuring Kerberos 4, see the *SP Administration Guide* (GC23–3897).

Using NIM to install clients configured with Kerberos 5 authentication:

In AIX 4.3.3 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 uses the following 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, the authentication methods on the NIM master should be set for both Kerberos 5 and Standard UNIX. 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, 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.

Using NIM to install clients with NIM resources that are exported with Kerberos authentication:

You can install NIM clients with NIM resources that are set with Kerberos security export.

This method provides added protection for NIM resources by preventing access from unacceptable hosts. To use this authentication method, the NIM master must be configured to be the Kerberos server.

Do the following:

1. Set up and configure the Kerberos server by using one of the following methods.

   Note: To avoid a base image installation failure, you must run one of the following commands.
   - If the NIM master is not configured as a Kerberos server, use the sample script that NIM provides by running the following command:

     ```
     /usr/samples/nim/krb5/config_rpcsec_server –u <user> -p <password>
     ```

     The **config_rpcsec_server** script runs the **/usr/lpp/bos.sysmgt/nim/methods/nimcrypt –u <*user*> -p <*password*>** command to setup the credentials for Kerberos authentication.
   - If the NIM master is configured as a Kerberos server, run the **nimcrypt** command:

     ```
     /usr/lpp/bos.sysmgt/nim/methods/nimcrypt –u <user> -p <password>
     ```
2. Set the **nfs_domain** attribute for the nim master by using one of the following methods.
   - Run the following command from the command line:

     ```
     nim –o change –a nfs_domain="austin.ibm.com" master
     ```
   - Use the following SMIT fastpath command:

     ```
     fastpath smitty nim_global_nfs
     ```
3. Set the NIM resources attributes for **nfs_sec** to krb5 and **nfs_vers** to 4 as follows:

   ```
   nim –o change –a nfs_sec=krb5 –a nfs_vers=4 <resource_object>
   ```

   Note: Setting `nfs_sec=krb5` for the SPOT resource is not supported for the install environment.

After the **nfs_sec** and **nfs_vers** attributes are set for the NIM resources and a NIM network installation is initialized, NIM uses NFS to export the location for the resource set with krb5. The client uses Kerberos authentication and mounts NIM resources over Kerberos security.

Installing a client with a Kerberos protected mount is only supported for NIM installations where `source=rte` or `source=mksysb`. A Kerberos installation will only work for NIM resources that reside on the NIM master. After a client authenticates with the Kerberos server, there is usually a time lease for the exported location to be active. This time lease defaults to 24 hours. If an installation exceeds 24 hours because of a system or network error, the installation will hang. If a hang occurs, troubleshoot the installation and restart the installation process by rebooting the client to network boot. The time lease can also be extended.

Using NIM to install clients configured with SSL authentication

NIM can be used to install machines in an RS/6000 environment configured for SSL authentication.

Clients configured for SSL authentication must use the NIM Service Handler (NIMSH) for handling NIM master push operations. For more information about NIMSH, see “Using the NIM service handler for client communication” on page 156.

You can install and configure the OpenSSL cryptographic software using the NIM command options. Scripts are provided for configuring OpenSSL in the NIM environment, and you can use these without any modifications. The scripts are installed as part of the `bos.sysmgt.nim.client` fileset and located in the `/usr/samples/nim/ssl` directory. The scripts are used to define SSL keys and certificates for NIM SSL usage.

Because NIM masters can support a large system environment, it is necessary to impose a hierarchy on SSL certificate and key storage structure. During NIM setup, the following directory structure is created:

/ssl_nimsh
: SSL parent directory for NIM

/ssl_nimsh/configs
: Contains scripts used to configure SSL in NIM

/ssl_nimsh/certs
Contains SSL certificates used during host authentication

/ssl_nimsh/keys
Contains SSL keys used during SSL protocol communication

The NIM SSL directory structure is considered static and you should not modify it. To change SSL certificate options, you can modify the following configuration scripts:

SSL_root.cnf
Generates Certificate Authority key for signing certificates

SSL_server.cnf
Generates the NIM master's certificate for distributing to clients

SSL_client.cnf
Generates the NIM master's local certificate for authenticating

Note: You should configure NIM SSL using default settings prior to modifying the configuration scripts. To verify changes, a certificate viewer script called **certview** is located in the `/usr/samples/nim/ssl` directory. For more information about **certview**, see "Using the certificate viewing file" on page 286.

For more information on installing and configuring OpenSSL in NIM, see the **nimconfig** command and **nimclient** command.

Using NIM to install clients configured with SSL authentication using SMIT:

Follow this procedure for using NIM to install clients configured with SSL authentication using SMIT.

To configure the NIM environment for SSL authentication, complete the following steps:
1. Type the fast path `smitty nim_ssl` on the NIM master.
2. Select **enabled** as the option for **Enabling Cryptographic Authentication**.
3. If OpenSSL is not installed on the client, select **yes** as the option for **Installing Secure Socket Layer Software**.
4. If OpenSSL is selected for installation, specify the absolute path for the `installp` package or select the **lpp_source** resource that contains the OpenSSL `installp` package.

Using NIM to install clients configured with SSL authentication from the command line:

Follow this procedure to configure the NIM environment for SSL authentication from the command line.
1. If OpenSSL is installed on the NIM master, type:
   ```
   # nimconfig -c
   ```
2. If OpenSSL is not installed on the NIM master, complete the following steps:
   - Locate the AIX Toolbox for Linux Applications media.
   - Install the OpenSSL RPM package using **geninstall**. For additional information on using **geninstall**, see Add Open Source Applications to Your AIX System.
   - After OpenSSL is installed on the NIM master, type:
     ```
     # nimconfig -c
     ```

Troubleshooting NIM OpenSSL:

A description of troubleshooting an error installing the OpenSSL package.

Problem

The NIM installation fails because it cannot find the `libssl.a` file. The `libssl.a` file is part of the OpenSSL package.

Solution

The error is a result of the mismatch between the AIX version of OpenSSL versus the RedHat Package Manager (RPM) version of OpenSSL. You should remove the AIX fileset and install the RPM version of OpenSSL on the NIM Client system.

Use the following information to resolve the problem.

OpenSSH is based on client and server architecture. OpenSSH runs the **sshd** daemon process on the AIX host and waits for the connection from clients. OpenSSH supports public-key and private-key pairs for authentication and encryption of access to ensure secure network connections and host-based authentication.

To download the latest **installp** format packages for the AIX operating system, go to the AIX Web Download Pack Programs website.

The following information explains how to install and configure OpenSSH on a system running the AIX operating system.

The OpenSSH software is shipped on the AIX base media. The `installp` packages include the man pages and the translated message filesets.

The following OpenSSH binary files are installed as a result of the preceding procedure:

scp A file copy program that is similar to the remote copy (rcp) file.

sftp A program similar to FTP that works over SSH1 and SSH2 protocol

sftp-server
The SFTP server subsystem, which is started automatically by the **sshd** daemon

ssh This is similar to the rlogin and rsh client programs

ssh-add
A tool that adds keys to the**ssh-agent** command

ssh-agent
An agent that can store private keys

ssh-keygen
A key generation tool

ssh-keyscan
A utility for gathering public host keys from a number of hosts

ssh-keysign
A utility for host-based authentication

ssh-rand-helper
A program used by OpenSSH to gather random numbers

Note: It is used only on AIX 5.1 installations.

sshd A daemon that permits you to log in

The following general information pertains to OpenSSH:

- The `/etc/ssh` directory contains the **sshd** daemon and the configuration files for the **ssh** client command.

- The /usr/openssh directory contains the readme file and the original OpenSSH open source license information. This directory also contains the **ssh** protocol and the Kerberos license information.
- The **sshd** daemon is under AIX SRC control. You can start, stop, and view the status of the daemon by issuing the following commands:

Command	Alternative
startsrc -s sshd	startsrc -g ssh (group)
stopsrc -s sshd	stopsrc -g ssh
lssrc -s sshd	lssrc -s ssh

  You can also start and stop the daemon by issuing one of the following commands:
  - /etc/rc.d/rc2.d/Ksshd start
  - /etc/rc.d/rc2.d/Ssshd start

    /etc/rc.d/rc2.d/Ksshd stop
  - /etc/rc.d/rc2.d/Ssshd stop
- When the OpenSSH server fileset is installed, an entry is added to the /etc/rc.d/rc2.d directory. An entry is in the inittab file to start run-level 2 processes (l2:2:wait:/etc/rc.d/rc 2) so that the **sshd** daemon will start automatically at boot time. To prevent the daemon from starting at boot time, remove the /etc/rc.d/rc2.d/Ksshd and /etc/rc.d/rc2.d/Ssshd files.
- OpenSSH software logs information to the SYSLOG log.
- The IBM Redbooks® publication, *Managing AIX Server Farms*, provides information about configuring OpenSSH in the AIX environment and is available in the IBM Redbooks.
- OpenSSH supports long user names of 256 bytes, the same as the AIX operating system.
- Some keywords, such as AllowUsers, DenyUsers, AllowGroups, and DenyGroups, are not available by default in the ssh_config file or the sshd_config file. You must add these keywords to the configuration files to use them.

Related information:

mkuser

OpenSSH

Get the latest version of OpenSSH for AIX

Managing AIX Server Farms Redbooks

Verifying installation with the lppchk operation

When investigating functional problems in software, you can use the **lppchk** operation to check the integrity of installed software. You can perform this operation from the SMIT, or the command line.

Verifying installation with the lppchk operation using SMIT:

Follow this procedure for verifying installation with the lppchk operation using 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.

Verifying installation with the lppchk operation from the command line:

Follow this procedure for verifying installation with the lppchk operation 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 with the * wildcard 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:

Item	Description
-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* equals 1 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
```

Performing a network installation of an IBM Power Systems over a virtual I/O Ethernet adapter

To perform a network installation on an IBM Power Systems server partition over a virtual I/O Ethernet adapter, the NIM master must be configured to receive packets from the partition adapter's default virtual local area network (VLAN).

One of the following configurations must exist:

- The master has a virtual I/O Ethernet adapter configured to receive packets from the partition's default VLAN, if the master is also a partition.
- A gateway exists that can route packets between the master's interface and the partition's default VLAN.
- The master has a VLAN interface associated with a physical Ethernet adapter that is configured to receive packets from the partition's default VLAN through the I/O server, if the IBM Power Systems server has an I/O server partition.

If you are performing a broadcast bootp installation, then you must have either the first or the third configuration. For additional information on configuring and using VLANs, see TCP/IP local area network adapter cards.

Setting default paging space during BOS installation through NIM

In AIX 4.3 or later, default paging space is set by the BOS installation process when installing through NIM.

Default paging space is set by the BOS installation process, 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 not a **mksysb** image.
- 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`
- The source of the BOS image is a **SPOT**, and 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 (Common Desktop Environment) is installed, **recommmended_ps** =
  - amount of **RAM** is less than 32 MB, then **recommended_ps** = 3 * **RAM**
  - amount of **RAM** is 32 MB or more, then **recommended_ps** = **RAM** + 64 MB
- IF CDE (Common Desktop Environment) is not installed, **recommmended_ps** =
  - amount of **RAM** is less than 32 MB, then **recommended_ps** = 2 * **RAM**
  - amount of **RAM** is 32 MB or more, then **recommended_ps** = **RAM** + 32 MB

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

Setting up NIM networks

When the NIM master is configured, the network associated with the master is automatically defined in the NIM environment. It is necessary only to define additional NIM networks if clients reside on other local area networks or subnets.

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.

Supported NIM network types

You can use these network types to support NIM.

- Ethernet
- Standard Ethernet
- IEEE 802.3 Ethernet
- Token-Ring
- FDDI
- ATM
- Generic
- HFI

Network boot support is provided for Ethernet, Token-Ring, and FDDI. Unlike other network adapters, ATM adapters cannot be used to boot a machine. Therefore, installing a machine over an ATM network requires special processing. See “Using NIM with ATM networks” on page 139. 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, non-booting operations, such as **cust** and **maint**, are allowed. Diskless and dataless clients cannot be associated with Generic networks, because 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:

Item	Description
-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."
-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**, **hfi**, and **generic**.

The following attributes are optional:

Item	Description
-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 187.
-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 "Defining NIM routes" on page 187, "Establishing a default NIM route between networks" on page 189, and "Establishing a static NIM route between networks" on page 190.
-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 108.

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
```

Defining NIM routes

NIM uses routing information internally 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 permit resources that are distributed throughout a NIM environment to be more easily accessed by any client in the NIM environment.

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.

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.

Defining a heterogeneous network

The NIM feature enables NIM to model networks consisting of different data-link protocol segments.

Networks consisting of different data-link protocol segments 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.

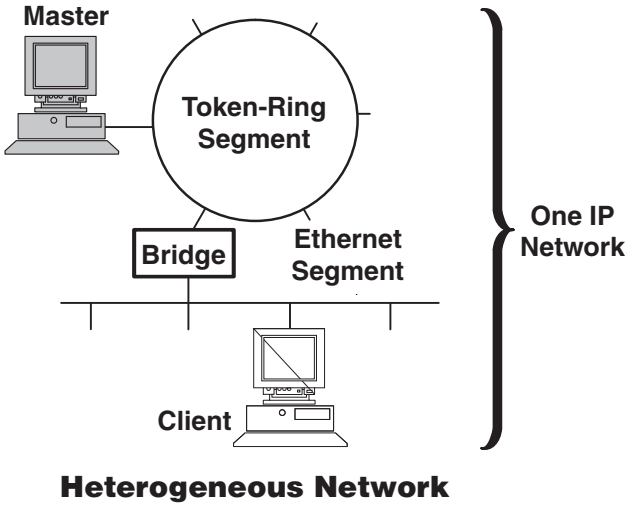

Figure 1. Heterogeneous Network. This illustration shows a single IP network in which the master server uses its token-ring connection and a bridge to communicate with its client on an Ethernet segment.

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>
```

The **other_net_type** attribute requires a sequence number 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

You can add a network type to a NIM network using SMIT, or the command line.

Adding another network type to a NIM network using SMIT:

Follow this procedure for adding another type of network to a NIM network using 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.

Adding another network type to a NIM network from the command line:

Follow this procedure for adding another network type to a NIM network from the command line.

To define a NIM network, enter:

```
nim -o change -a other_net_type<SequenceNumber=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
```

Establishing a default NIM route between networks

You can create default NIM routes for two Networks (for example, `Network1` and `Network3`).

Establishing a default NIM route between networks using SMIT:

Follow this procedure for establishing a default NIM route between networks using SMIT.

1. 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.

Establishing a default NIM route between networks from the command line:

Follow this procedure to establish default NIM routes between networks 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

You can create a static NIM route between two networks (for example, Network1 and Network3) using the SMIT, or the command line.

Establishing a static NIM route between networks using SMIT:

Follow this procedure for establishing a static NIM route between networks using SMIT.

1. 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.

Establishing a static NIM route between networks from the command line:

Follow this procedure for establishing a static NIM route between networks 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 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    =  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
```

Booting with NIM

Review the different ways you can use boot images with NIM.

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/DVD-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.

After successfully booting and defining the console, the System Maintenance menu is displayed. The maintenance menu options and their descriptions are described below.

Item	Description
Access a Root Volume Group	Use this option to activate the root volume group with or without mounting the file system, and start a maintenance shell. When the file systems are mounted, you have access to the full set of commands within the shell. **Note:** Once you access a root volume group, you will not be able to return to the Base Operating System Installation menus without rebooting.
Copy a System Dump to Removable Media	Use this option to copy a previous system dump to external media.
Access Advanced Maintenance Function	Use this option to start a maintenance shell with a limited set of commands. To return to the Maintenance menu, type **exit**.
Erase Disks	Use this option to select one or more disks for erasure. Next, you can select the number of patterns to write on the disk, from a set of choices. To return to the previous menus, type **99**.
Configure Network Disks (iSCSI)	This option puts you into a SMIT interface to configure an iSCSI disk. To return to the Base Operating System Installation menus, use the SMIT **F10** exit key.
Select Storage Adapters	Use this option to select the disk adapter for the installation destination disk. Only disks attached to the system through this adapter are displayed. The name and location code for the disk adapter are also displayed. The location code indicates the slot where the disk adapter is connected. To return to the previous menus, type **99**.

Booting in maintenance mode using SMIT:

Follow these procedures for booting in maintenance mode using SMIT.

Initiating the maint_boot operation from the client:

Follow this procedure for 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:

Follow this procedure for 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.

Booting in maintenance mode from the command line:

Follow these procedures for booting in maintenance mode 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. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.

Performing boot diagnostics on NIM clients

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/DVD-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.

Booting diagnostics using SMIT:

Follow these procedures for performing the **diag** operation from the master and client using SMIT.

Initiating the diag operation from the client:

Follow this procedure to initiate 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:

Follow this procedure to initiate 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.

Booting diagnostics from the command line:

Follow this procedure for performing the **diag** operation from the master and client.

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. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.

Loading diagnostics without the diag operation:

In addition to the procedure using the diag operation, 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.

Initializing and booting a diskless or dataless machine

Use this procedure to configure and boot a machine as a diskless or dataless client in the NIM environment.

Prerequisites

- 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" on page 144.
- 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 147 procedure.

Initializing and booting a diskless or dataless machine using SMIT:

Follow this procedure for initializing and booting a diskless or dataless machine using 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, boot the client machine over the network. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.

   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.

Initializing and booting a diskless or dataless machine from the command line:

Follow this procedure for initializing and booting a diskless or dataless machine from the command line.

1. To initialize the client resources for diskless clients, complete one of the following depending on which resource is used:
   - If a **root** resource is used, enter the following on the NIM master:
     ```
     nim -o dkls_init -a spot=SPOTName -a root=RootName \
     -a dump=DumpName -a paging=PagingName ClientName
     ```
   - If a **shared_root** resource is used, enter the following on the NIM master:
     ```
     nim -o dkls_init -a spot=SPOTName -a shared_root=SharedRootName \
     -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 "Using the NIM dkls_init operation" on page 271 and "Using the NIM dtls_init operation" on page 272.
3. After completion of the initialization operation, boot the client machine over the network. If you are booting from a network device, follow the procedures in your hardware documentation to perform the network boot.

   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.

Booting over a router on an FDDI

Boot over a router on an Fiber Distributed Data Interface (FDDI) only if the router supports all-route broadcast.

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

Administering NIM

You can use NIM to complete different types of task including backing up and restoring a NIM database.

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.

Recovering the /etc/niminfo file from the command line:

Follow this procedure for recovering the /etc/niminfo file from the command line.

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

```
nimconfig -r
```

To rebuild 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

You can back up the NIM database using SMIT, or the command line.

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.

Backing up the NIM database using SMIT:

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

Backing up the NIM database from the command line:

Follow this procedure to back up the NIM database from the command line.

The following NIM files must be saved for backup:

- `/etc/niminfo`
- `/etc/objrepos/nim_attr`
- `/etc/objrepos/nim_attr.vc`
- `/etc/objrepos/nim_object`
- `/etc/objrepos/nim_object.vc`
- `/etc/NIM.level`
- `/etc/niminfo`
- `/etc/NIM.primary.cpuid`

Best practice: Back up the NIM database to the `/home` directory or a user-created file system.

To back up the database, enter the following command:

```
/usr/lpp/bos.sysmgt/nim/methods/m_backup_db backup_file_name_and_location
```

For example,

```
# /usr/lpp/bos.sysmgt/nim/methods/m_backup_db /home/backup.070915
a ./etc/objrepos/nim_attr 48 blocks.
a ./etc/objrepos/nim_attr.vc 144 blocks.
a ./etc/objrepos/nim_object 8 blocks.
a ./etc/objrepos/nim_object.vc 40 blocks.
a ./etc/NIM.level 1 blocks.
a ./etc/niminfo 1 blocks.
a ./etc/NIM.primary.cpuid 1 blocks.
```

Restoring the NIM database and activating the NIM master

You can restore the NIM database and activate the NIM master using the SMIT, or the command line.

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

Restoring the NIM database and activating the NIM master using SMIT:

Follow this procedure for restoring the NIM database and activating the NIM master using SMIT.

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

Restoring the NIM database and activating the NIM master from the command line:

Follow this procedure for restoring the NIM database and activating the NIM master from the command line.

To restore the NIM database and activate the NIM master, enter the following command:

```
/usr/lpp/bos.sysmgt/nim/methods/m_restore_db backup_file
```

For example, if you backed up the NIM database by using the following command:

```
# /usr/lpp/bos.sysmgt/nim/methods/m_backup_db /home/backup.070915
```

Restore the NIM database by using the following command:

```
# /usr/lpp/bos.sysmgt/nim/methods/m_restore_db /home/backup.070915
```

Related tasks:

"Backing up the NIM database from the command line" on page 195
Follow this procedure to back up the NIM database from the command line.

Removing machines from the NIM environment

You can remove a machine from the NIM environment 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.

Removing machines from the NIM environment using SMIT:

Follow this procedure for removing a machine from the NIM environment using SMIT.

1. Enter the **smit nim_rmmac** fast path.
2. Select the machine to remove.

3. In the displayed dialog fields, accept the defaults.

Removing machines from the NIM environment using the command line:

Follow this procedure for removing machines from the NIM environment using the command line.

Enter:

```
nim -o remove MachineName
```

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

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.

Removing members from machine groups using SMIT:

Follow this procedure for removing members from machine groups using 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.

Removing members from machine groups from the command line:

Follow this procedure for removing members from machine groups 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 the machine `Standalone2`, and add the machine `Standalone4` to the group `MacGrp1`, enter:

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

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.

Preventing machines from adding themselves as clients using SMIT:

Use this information to change the option to allow machines to add themselves to a NIM environment as clients.

Type the SMIT fast path:

```
smit nim_client_reg
```

Preventing machines from adding themselves as clients from the command line:

Use this information to set from the command line whether machines can add themselves as clients in a NIM environment.

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
```

Adding mount options to NIM clients

You can add mount options to NIM clients. The mount options can be used when resources are mounted on the client.

To set the mount options, use the following command:

```
nim -o change -a mount_opts=MountOptions MachineName
```

Note: If you are using Network File System (NFS) version 4, the **-o** flag cannot be used with the **mount_opts** attribute.

Examples

1. To add mount options with NFS version 3, enter the following command:

   ```
   nim -o change -a mount_opts="-o intr,vers=3,proto=udp" client1
   ```

2. To add mount options with NFS version 4, enter the following command:

   ```
   nim -o change -a mount_opts="proto=udp" client1
   ```

Exporting NIM resources globally

NIM resources can be exported globally using SMIT, or the command line interface.

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 exceeded in the files, and it may also negatively affect NIM performance as the files are locked and modified for each resource allocation or 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.

Exporting NIM resources globally using SMIT:

Use this information to export NIM resources globally using SMIT.

To manage global exporting of NIM resources from the SMIT interface, type the SMIT fast path:

```
smit nim_global_export
```

Exporting NIM resources globally 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
```

Do not change the enablement and disablement of global exports when there are resources allocated to clients because 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. If resources are currently allocated to clients, the **nim** command will fail to change the **global_export** value.

Enabling NIM alternate master support

Using this procedure, you can set up an alternate NIM master in your environment, synchronize the NIM database between masters, and takeover control of clients between masters.

Preparing to enable NIM alternate master support:

Before you create an alternate master for your NIM environment, you should already have a primary NIM master configured. Both masters must be at the same level of AIX.

For instructions about configuring a NIM master, see Configuring the NIM Master and Creating Basic Installation Resources.

Initializing the alternate master:

You can initialize the alternate master in SMIT or from a command line.

In this scenario, master A is already configured as a NIM master, and master B will be initialized as an alternate master.

Initializing the alternate master using SMIT:

Follow this procedure for initializing the alternate master using SMIT.

In this scenario, master A is already configured as a NIM master, and master B will be initialized as an alternate master.

1. Insert the *AIX Volume 1* media into the appropriate drive of the designated alternate master system (master B).
2. Enter the SMIT **install_latest** fast path, to install the `bos.sysmgt.nim.master` fileset.
3. Using the **LIST** option, select **/dev/cd0** for the **INPUT** device or directory for software.
4. Specify **bos.sysmgt.nim.master** as the SOFTWARE to install.
5. Accept the default values for all other fields on this display. After successful completion of this installation, exit SMIT.
6. To initialize the alternate master (master B) with master A, enter the **smit niminit_altmstr** fast path on the master B system.
7. Type the name of master B in the **This Machine Name** field.
8. Using the **LIST** option, select the Primary Network Interface for master B.
9. Type the host name of master A in the **Host Name of Master with which to Initialize**field.
10. Change any other fields as necessary, and press Enter.
11. On master A, repeat the process by using the **smit niminit_altmstr** fast path to register master A with master B. You will need to type the name of master A in the **Machine** field and the host name of master B in the **Host Name of Master** with which to Initialize.

Consider the following items when you initialize the alternate master from SMIT:

- The **niminit** command creates an **alternate_master** object for the registering system. In this example, master B is defined as an **alternate_master** object on master A when master B is registered with master A.
- The **niminit** command configures the **alternate_master** object as a NIM master if it is not already configured as one.
- The **niminit** command notifies the master that the system is registering with remote access permissions through the **nimsh** shell. In this example, when master B is registered, master B gives master A remote access permissions.
- After an alternate master has been added to the NIM environment, clients initialize themselves again to recognize the alternate master. Initializing again gives the alternate master remote access to the clients either through the **rsh** or **nimsh** shells. After clients have initialized themselves again, their **sync_required** attribute is set to **no**, indicating that they recognize the alternate master.

Initializing the alternate master from the command line:

Follow this procedure for initializing the alternate master from the command line.

In this scenario, master A is already configured as a NIM master, and master B will be initialized as an alternate master.

1. Insert the *AIX Volume 1* media into the appropriate drive of the designated alternate master system (master B).
2. To install the `bos.sysmgt.nim.master` fileset from the disk, enter the following command on the master B system.

   ```
   # installp -agXd /dev/cd0 bos.sysmgt.nim.master
   ```
3. To initialize the alternate master (master B) with master A, enter the following command on the master B system.

   ```
   # niminit -a is_alternate=yes -a attr1=value1 \
   -a attr2=value2 \
   ...
   ```

   Assume the following data to initialize the alternate master (master B) with the existing NIM master (master A) :

   ```
   alternate master host name = masterb
   master host name with which to register = mastera
   primary network interface = en0
   cable type = N/A
   platform = chrp
   ```

   With the previous assumptions, enter the following command on master B system:

   ```
   # niminit -a is_alternate=yes -a master=mastera -a pif_name=en0 \
   -a cable_type1=N/A -a platform=chrp -a name=masterb
   ```

   For additional attribute information, see the **niminit** command.
4. Register master A with master B by using the **niminit** command. For example, enter the following command on the master A system:

   ```
   # niminit -a is_alternate=yes -a master=masterb -a pif_name=en0 \
   -a cable_type1=N/A -a platform=chrp -a name=mastera
   ```

Synchronizing the alternate master's NIM database:

You can synchronize the NIM database for the alternate master using SMIT or from the command line.

In this scenario, master A is configured as a NIM master and has objects, such as clients and resources, defined. Master B is initialized as an alternate master, but its database does not match that of master A. You can use the **sync** operation to synchronize the NIM database on master B with master A's database. The **sync** operation backs up master A's database, restores it onto master B, and then ensures that all the object definitions are consistent.

You should consider the following issues when synchronizing the alternate master's NIM database:

- The resources served by master A are removed from the database when it is restored on master B.
- Object definitions are reset when the database is restored on master B.
- After the database is restored on master B, master B does not control any NIM objects until you perform the **takeover** operation. As a result, master B can not run any NIM operations to any objects in its database.

Synchronizing the alternate master's NIM database using SMIT:

Follow this procedure for synchronizing the alternate master's NIM database by using SMIT.

1. On master A, enter the**smit nim_altmstr** fast path.
2. Select **Synchronize an Alternate Master's NIM database**.
3. Type the name of the NIM object for master B.
4. Select **yes** for the force option if master B is configured as a NIM master.
5. Select **yes** for the replicate option if the resources are replicated onto the alternate master.
6. Select **yes** for the **Reset NIM Client to Alternate Master** option so that NIM clients are aware of the alternate master.

Synchronizing the alternate master's NIM database from the command line:

Follow this procedure to synchronize the alternate master's NIM database from the command line.

To synchronize master B's database with master A's, enter the following on master A:

```
# nim -o sync masterb
```

To synchronize master B's database with master A's, and also to replicate the resources served by master A:

```
# nim -o sync -a replicate=yes masterb
```

To synchronize master B's database with master A's, replicate the resources served by master A, and rebuild the NIM clients list in `/etc/niminfo` to be aware of the alternate master:

```
# nim -o sync -a replicate=yes -a reset_clients=yes masterb
```

Notes:

- You must use the force option to overwrite the existing database that was created by the **niminit** command.
- Resources are replicated only if they are not present in the appropriate file system locations of the alternate master.

For example:

```
# nim -Fo sync masterb
```

The replicate option can be used along with the force option. For example:

```
# nim -Fo sync -a replicate=yes masterb
```

The reset_clients option can be used along with the force option. For example:

```
# nim -Fo sync -a reset_clients=yes masterb
```

Taking control of the NIM environment:

You can take control of the NIM environment using SMIT or from the command line.

In this example, master B is initialized as an alternate master and has had its NIM database synchronized with master A. Master B takes control of the objects in the NIM environment.

You should consider the following issues when taking control of the NIM environment:

- If you perform this operation while master A is running, and master A has a network connection to master B, the database on master A is updated to reflect the change in masters. You should reset the clients on master A before running this operation. This operation prints warnings for any clients on master A that are currently set up to run NIM operations. This operation will not reset those clients to complete successfully.
- If master B is unable to update master A during the takeover operation, master A should have its database synchronized with master B once it is running.
- This operation updates the current master of each client by running a remote command on the clients. Clients that are unavailable to switch masters are displayed. When the client is available, initialized it with master B or perform the takeover operation again from master B.

Taking control of the NIM environment using SMIT:

Follow this procedure for taking control of the NIM environment using SMIT.

1. Enter the **smit nim_altmstr** fast path on master B and select **Takeover control of NIM clients from an Alternate Master**.
2. Type in the name of the NIM object for master A.

Taking control of the NIM environment from the command line:

Follow this procedure for taking control of the NIM environment from the command line.

To have master B take control of the NIM environment, enter the following on master B:

```
# nim -o takeover mastera
```

Master A can retake control of the NIM environment by running the **takeover** command with master B as the target.

Removing an alternate master from the NIM environment:

You can remove an alternate master from the NIM environment using SMIT or from the command line.

In this example, master B is removed from the NIM environment. Master A should be in control of the NIM environment prior to removing master B.

You should consider the following issues when removing an alternate master from the NIM environment:

- Clients re-initialize themselves with master A after removing master B from the environment. Re-initializing updates the `niminfo` files and remote access permissions.
- You can unconfigure Master B by running the **unconfig** operation locally on master B.

Removing an alternate master from the NIM environment using SMIT:

Follow this procedure for removing an alternate master from the NIM environment using SMIT.

1. On master A, enter the **smit nim_altmstr** fast path and select **Remove an Alternate Master**.
2. Select the NIM name of master B.

Removing an alternate master from the NIM environment from the command line:

Use this command to remove an alternate master from the NIM environment.

To remove master B from the NIM environment, enter the following command on master A:

```
# nim -o remove masterb
```

Configuring SSL authentication on an alternate master:

Use this process to configure SSL authentication on an alternate master.

You can configure SSL communication on an alternate master. The alternate master will need to install the `openssl.base` fileset. With the SSL fileset installed on the alternate master, the NIM master must be configured with SSL authentication using the article topic "Using NIM to install clients configured with SSL authentication from the command line".

Follow these procedures to configure SSL authentication for the alternate master from the command line.

- If OpenSSL is installed on the NIM alternate master, to configure SSL on the alternate master, type:

  ```
  # nimconfig -c
  ```

- If OpenSSL is installed on the NIM alternate master, to establish SSL communication with the NIM master, type:

  ```
  # nimclient -c
  ```

- If OpenSSL is installed on the NIM alternate master, to establish SSL communication from the alternate master with each of the NIM client, type on each of the NIM client, where `<alternate_master>` is the name of the alternate_master.

  ```
  # nimclient -o get_cert -a master_name=<alternate_master>
  ```

Migrating a NIM client to an IBM Power Systems server logical partition

The **nim_move_up** application allows you to easily migrate a back-level AIX system onto an logical partition (LPAR) residing on an IBM Power Systems server.

The system must meet the following requirements before you can run the **nim_move_up** application properly.

- NIM Master Requirements
  - A configured NIM master
  - Perl 5.6 or above
  - Openssh (obtainable from the Linux Toolbox media)
  - At least one stand-alone NIM client running AIX
  - AIX product media version, or equivalent `lpp_source` and `SPOT` NIM resources
- Server and resource requirements
- An IBM Power Systems server with sufficient hardware resources to support the target clients' equivalent IBM Power Systems configuration
- If virtual resources will be used to migrate the clients, an installed and configured Virtual I/O Server is required
- HMC controlling the IBM Power Systems server, along with sufficient privileges to start, stop, and create LPARs
- root user authority

This nim_move_up process requires no downtime on the part of the original client. In addition, **nim_move_up** is capable of migrating a client onto virtualized hardware, such as virtual disks, using the

Virtual I/O capabilities of the IBM Power Systems server. This migration process can be completed by the **nim_move_up** application in phases to allow more control over the process, or it can be completed all at once without any user interaction required.

With the **nim_move_up** application, you can use a NIM master and its clients as the starting point for a migration that produces the following hardware environment:

- The original NIM master
- LPARs on an IBM Power Systems server that correspond to the original NIM clients and are controlled by the NIM master
- HMC to control the LPARs on the IBM Power Systems servers, communicated with by the NIM master through SSH
- The original NIM clients

The **nim_move_up** migration process is completed in the following phases to allow more control over the process.

1. The *Create NIM Resources* phase creates the needed NIM resources to perform the migration steps if they don't already exist or are not provided beforehand.
2. The *Pre-migration Software Assessment* phase performs an assessment on each target client to determine what software is installed and can be migrated. Any software that is missing from the `lpp_source` will be added from the source of installation images that should be provided to nim_move_up.
3. The *Client Hardware and Utilization Data Gathering* phase gathers data about each target client's hardware resources and attempts to assess how much of those resources are utilized on average over a given amount of time.
4. The *IBM Power Systems Resource Availability Data Gathering and Client Resource Data Translation* phase searches the given managed system for available hardware resources. Uses the data gathered in the previous phase to create an equivalent LPAR configuration that utilizes the managed system's available resources. Creates the client LPARs with virtual I/O resources instead of physical I/O resources if **nim_move_up** was provided a Virtual I/O Server LPAR to work with. Creates the appropriate adapters and configuration on the Virtual I/O Server as they are needed.
5. The *Create System Backups of Target Clients* phase creates an installable image of each target client and its resources using the **mksysb** command.
6. The *Migrate Each System Backup* phase uses the nimadmin command to migrate the newly-created installable images to the new level of AIX.
7. The *Allocate NIM Resources to New LPARs* phase uses the network information provided to the nim_move_up application to create NIM standalone client objects for the new LPARs created in the *IBM Power Systems Resource Availability Data Gathering and Client Resource Data Translation* phase. Allocates the appropriate NIM resources and runs a **bos_inst pull** operation (i.e. NIM will not attempt to boot the client) on each NIM client.
8. The *Initiate Installation on LPARs* phase reboots each LPAR via the control host (HMC partition) and initiates the installation.

   Note: This phase ends when the installation begins. The actual progress of the installation is not monitored.
9. **Post-migration Software Assessment**

   Assesses the overall success of the migration after each installation, and reports on any software migration issues. It may be necessary to manually correct the errors reported for filesets that fail to migrate.
10. **Post-installation Customization**

Performs a NIM customization operation on each client with the values provided if an alternate lpp_source, fileset list, or customization script was provided to the **nim_move_up** application. This allows for the optional installation of additional software applications or for any additional customization that may be needed.

Migrating a NIM client to an IBM Power Systems server logical partition using SMIT:

The SMIT fastpath to the root menu of **nim_move_up** is `smitty nim_move_up`.

After all prerequisites needed to run the **nim_move_up** application have been met, **nim_move_up** performs the migration process in two steps: configuration and phase execution. You can run the **nim_move_up** allocation from SMIT by completing the following steps:

1. Enter `smitty nim_move_up_config`. The **Configure nim_move_up Input Values** panel opens.
2. Enter information in the required fields. This information is retained by the **nim_move_up** application, unless the application is reset. You can change this information at any time from the **Configure nim_move_up Input Values** panel.
3. To begin the actual migration process, enter `smitty nim_move_up_exec`. The **Execute nim_move_up Phases** panel opens.
4. Provide an appropriate answer to the option **Execute All Remaining Phases?** on the **Execute nim_move_up Phases** panel and press Enter.

You can use other panels to interact with the **nim_move_up application**, in addition to the **Configure nim_move_up Input Values** panel and the **Execute nim_move_up Phases** panel:

Display the Current Status of nim_move_up
Selecting this menu option is equivalent to running **nim_move_up** with the **-S** flag. The next phase to be executed and a listing of all the saved options are displayed.

Configure SSH Keys on Target HMC
This SMIT panel provides a simple interface to setting up SSH keys on the remote control host (HMC). Using this panel is the equivalent of using the **-K** command line option. Configuring SSH keys on the remote control host enables the unattended remote execution of commands from the NIM master.

Unconfigure nim_move_up
This SMIT panel provides an interface to unconfiguring the **nim_move_up** environment. Unconfiguring the environment removes all state information, including what phase to execute next, saved data files generated as a result of the execution of some phases, and all saved input values. Optionally, all NIM resources created through **nim_move_up** can also be removed. Using this panel is the equivalent of using the **-r** command line option.

Migrating a NIM client to an IBM Power Systemsserver logical partition using the command line:

Once all prerequisites needed to run the **nim_move_up** application have been met, **nim_move_up** performs the migration process in two steps: configuration and phase execution.

Command-Line Usage

nim_move_up {[-S] | [-K [-h *control_host*]] | [-r [-R]]} | { [-c *NIM_client*] [-i *target_ip*[-*ending_ip*]] [-s *subnet_mask*] [-g *gateway*] [-h *control_host*] [-m *managed_sys*] [-V *vio_server* [-e] [-D]] [-I *img_src*] [-l *resource_dir*] [-t *seconds*] [-p *loops*] [-j *nimadm_vg*] [-L *lpp_source*] [-U *spot*] [-B *bosinst_data*] [-E *exclude_files*] [-C *script_resource*] [-b *installp_bundle*] [-f *fix_bundle*] {{[-n] [-d]} | -O} [-q] }

Table 12. Required Flags

Flag	
-c *NIM_client*	Either a NIM standalone client (standalone object type), or a NIM machine group (mac_group object type). The indicated clients must be reachable via the network from the NIM master and must allow the NIM master to execute commands on them. If you specify a NIM machine group in this argument, they must all reside in the same NIM network. The clients will be the target machines that will be migrated onto equivalent LPARs on an IBM Power Systems server.
-i *target_ip*[*-ending_ip*]	The IP address that the new migrated client will be configured with after it is installed onto the IBM Power Systems server. If a NIM machine group is supplied to the **-c** option, a range of IP addresses must be supplied here and there must be enough addresses in the range to enumerate the amount of clients that will be migrated.
-s *subnet_mask*	The subnet mask that the clients will be configured with after the migration to the IBM Power Systems server.
-g *gateway*	The IP address of the default gateway that the clients will be configured with after the migration to the IBM Power Systems server.
-h *control_host*	The hostname or IP address of the HMC that is used for hardware control of the IBM Power Systems server that **nim_move_up** is to use.
-m *managed_sys*	The name of the managed system corresponding to the IBM Power Systems server as tracked by the HMC.
-I *img_src*	Path to the source of the installation images to be used to create the NIM resources needed to perform the migration and installation. This path can be a device, such as **dev/cd0** if using AIX product media, or a path to a location on the file system containing the installation images.
-l *resource_dir*	Path to a location on the file system that will contain any new NIM resources created through **nim_move_up**. The location should have enough space to accommodate an *lpp_source* and a spot unless existing resources were provided through the **-L** and **-U** options.

Table 13. Execution and Control Flags

Flag	
-S	Displays the status of the execution of the current phase or the next phase to be executed. All saved values are displayed as well. **nim_move_up** exits immediately after displaying the information. This flag cannot be used with any other options.
-n	Executes only the next phase of the **nim_move_up** migration process. **nim_move_up** will exit when the phase completes or fails. If you do not provide this flag, all the subsequent phases will be executed, and **nim_move_up** will exit when all the phases have executed, or one of them has failed.
-d	**nim_move_up** will execute in the background and return control of the terminal to the caller. The progress of **nim_move_up** can be tracked through the **-S** flag described above.
-q	Quiet mode. No output will be printed to the terminal, but will instead be kept in the logs. This flag has no effect if **nim_move_up** is being executed with the **-d** flag described above.
-O	Only save supplied values. **nim_move_up** will save values provided through other options and then exit without executing any phases. This flag cannot be used with any other of the execution or control flags.
-K	Configures SSH keys on the specified HMC to allow the unattended remote execution of commands from th NIM master without password prompts. This flag cannot be used with any other options except for the **-h** option.

Table 13. Execution and Control Flags (continued)

Flag	
-r	Unconfigures **nim_move_up**, which causes it to reset all its saved data, including saved options, phase-specific data, and current phase information. This operation must be executed if the migration process is to be started over for the migration of a new client or set of clients.
-R	Removes all NIM resources created by **nim_move_up** in addition to unconfiguring the environment. This flag can only be used with **-r**, described above.

Table 14. Optional Flags

Flag	
-V *vio_server*	LPAR name of a Virtual I/O Server residing on the IBM Power Systems server denoted through the **-m** flag described above.
-e	Forces the use of physical network adapters instead of shared ethernet adapters in creating the new LPAR on the IBM Power Systems server when a Virtual I/O Server LPAR has been specified. This flag is only valid when used with the **-V** option described above.
-D	Forces the use of physical storage controllers instead of virtual SCSI adapters in creating the new LPAR on the IBM Power Systems server when a Virtual I/O Server LPAR has been specified. This flag is only valid when used with the **-V** option described above.
-p *loops*	Number of times to execute system analysis tools on the target NIM clients in analyzing its resource utilization. The final resource utilization data will be the average of the values obtained from each loop and will be taken into account when determining the equivalent IBM Power Systems server resources from which the migrated LPAR will be derived. If you do not provide this option, it will default to 1 loop.
-t *seconds*	Number of seconds for which each loop runs. If you do not provide this option, it will default to 10 seconds.
-j *nimadm_vg*	The volume group to be used by the underlying **nimadm** call for data caching. If this option is not provided, the default value will be rootvg.
-L *lpp_source*	An existing *lpp_source* NIM resource to whose AIX level the target clients will be migrated to. If this option is not provided, **nim_move_up** will attempt to create a new *lpp_source* from the installation image source provided through the **-I** option, described above.
-U *spot*	An existing spot NIM resource that will be used in the migration and installation of the clients. If this option is not provided, a new spot will be created from the provided *lpp_source* NIM resource (see the **-L** and **-I** options above).
-B *bosinst_data*	An existing *bosinst_data* NIM resource that will be used by **nim_move_up** to install the new clients onto the IBM Power Systems server LPARs. If this option is not provided, **nim_move_up** will generate a *bosinst_data* resource with default unattended installation values.
-E *exclude_files*	An existing *exclude_files* NIM resource that **nim_move_up** will use when creating a **mksysb** of the original clients. If this option is not provided, **nim_move_up** will generate an *exclude_files* resource that will exclude the contents of **/tmp** from the backup.
-C *script_resource*	An existing script NIM resource that, if provided, **nim_move_up** will execute in phase 10 (Post-installation Customization) on all of the newly migrated LPARs.

Table 14. Optional Flags (continued)

Flag	
-b *installp_bundle*	An existing *installp_bundle* NIM resource whose software will be installed on each of the new migrated LPARs in phase 10 (Post-installation Customization) if the option is provided to **nim_move_up**.
-f *fix_bundle*	An existing *fix_bundle* NIM resource whose APARs will be installed on each of the new migrated LPARs in phase 10 (Post-installation Customization) if the option is provided to **nim_move_up**.

Example:

To configure the **nim_move_up** application with the required options and to start the first phase of the migration process, you would enter the following:

```
nim_move_up -c client1 -i 192.168.1.100 -s 255.255.255.0 -g 192.168.1.1 -h hmc1.mydomain.com -m
my-p5 -l /big/dir -I /dev/cd0 -n
```

where

- *-c client1* is a NIM standalone client reachable via the network from the NIM master
- *-i 192.168.1.100* is the IP address that the new migrated client will be configured with after it is installed onto theIBM Power Systems server
- *-s 255.255.255.0* is the subnet mask that the clients will be configured with after the migration to the IBM Power Systems server
- *-g 192.168.1.1* is the IP address of the default gateway that the clients will be configured with after the migration to the IBM Power Systems server
- *-h hmc1.mydomain.com* is the hostname or IP address of the HMC that is used for hardware control of theIBM Power Systems server to be used by the nim_move_up application
- *-m my-p5* is the name of the managed system corresponding to the IBM Power Systems server as tracked by the HMC
- *-l /big/dir* is the path to a location on the file system that will contain any new NIM resources created by the nim_move_up application
- *-I /dev/cd0* is the path to the source of the installation images to be used to create the NIM resources needed to perform the migration and installation
- *-n* begins the next phase of the migration process.

Then, to execute all remaining phases of the migration process in the background and save your agreement to accept all licenses, you would enter

```
nim_move_up -Y -d
```

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. You can view these logs from the SMIT, or the command line.

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:

Item	Description
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.

Viewing installation, configuration, and boot logs using SMIT:

Follow this procedure for viewing installation, configuration, and boot logs using 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).

Viewing installation, configuration, and boot logs from the command line:

Follow this procedure for viewing installation, configuration, and boot logs from the command line.

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

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

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

Disabling master push permissions in the NIM environment

The NIM master must have push permissions to perform push operations on the NIM clients.

You can disable the NIM master's push permissions using SMIT, or the command line.

Disabling master push permissions using SMIT:

You can use the **smit nim_perms** fast path to disable the master push permissions.

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

Disabling master push permissions from the command line:

You can disable and re-enabling the master push permissions 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 the NIM state

To return a machine to the **ready** state, use the NIM **reset** operation.

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.

Resetting the NIM state using SMIT:

Follow this procedure for resetting the NIM state using 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**.

Resetting the NIM state from the command line:

Follow this procedure for resetting the NIM state 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.

Rebuilding network boot images for a SPOT

You can rebuild network boot images for a SPOT using the SMIT, or the command line.

Rebuilding network boot images for a SPOT using SMIT:

Follow this procedure for rebuilding network boot images for a SPOT using 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**.

Rebuilding network boot images for a SPOT from the command line:

Follow this procedure to force the rebuild of the boot images for a SPOT from the command line.

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 141.

Migrating diskless and dataless clients and NIM SPOTS

Migration to a new release of AIX is not 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.

Performing advanced NIM installation tasks

You can perform many advanced NIM installation tasks using the NIM interface, the System Management Interface Tool (SMIT), or the command line.

Defining machine groups:

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.

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 with the same architecture.

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

Defining a machine group using SMIT:

Follow this procedure for defining a machine group using 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.

Defining a machine group from the command line:

Follow this procedure for defining a machine group 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.

Adding new members to machine groups using SMIT:

Follow this procedure for adding new members to machine groups using 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.

Adding new members to machine groups from the command line:

Follow this procedure for adding new members to machine groups 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.

Including and excluding group members from operations on the group:

Group members may be included or excluded by using the 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.

Including and excluding a group member from operations on the group using SMIT:

Follow this procedure for including and excluding a group member from operations on the group using 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.

Including and excluding a group member from operations on the group from the command line:

Follow this procedure for including and excluding a group member from operations on the group 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 `MacGrp1` machine group:

```
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 the nimdef command

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

The **nimdef** command also 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. For a sample definition file for the **nimdef** command, see "Using network installation files" on page 284.

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 the NIM **cust** operation.

This process will update all current SPOTs with the latest 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 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 install all devices on the installation media. For further details about the **cust** operation, see "Using the NIM cust operation" on page 268.

Managing NIM

You can manage Network Installation Management (NIM) using the NIM interface, System Management Interface Tool (SMIT), and the command line.

NFS client communication options management

AIX Network Installation Management (NIM) provides several options for network security and firewall enhancements.

The NIM Service Handler (NIMSH) provides you with several options for remote service authentication and limits the network socket selection of the service. NIMSH provides NIM users with a client configurable option for service authentication. Use Network File System (NFS) V4, which is part of NIM, to encrypt or secure network data on resource servers.

NFS V4 provides information-security functions:

Identification
: Establishes the identity of any users, hosts, or services

Authentication
: Confirms the identity of a user, host, or service

Authorization
: Controls what shared information each user or entity can access

The information-security functions in the network installation environment use NIM's object-oriented description of an install model. Resource objects in the NIM database must contain additional attributes for describing the security options required when accessing NIM resources through NFS V4.

NFS V4 host identification:

The NFS V4 server identifies client hosts using these methods.

Basic host identification

An NFS V4 server identifies client hosts by the IP address given in the Remote Procedure Call (RPC) packets. The NFS server turns this IP address into a host name using a host resolver, which gets its information from the Domain Name System (DNS) or the local `/etc/hosts` file.

Kerberos host identification

Kerberos authentication uses a unique identifier called a machine principal to identify hosts. The machine principal is established when configuring a host into a Kerberos realm. The machine principal name is the fully qualified host name prefixed with `host/` (for example, `host/jsblade00.austin.ibm.com`).

Kerberos can indirectly identify a host is through the NFS service principal (the identification of the NFS service running on the host). The service principal name is the fully qualified host name prefixed with `nfs/` (for example, `nfs/jsblade00.austin.ibm.com`).

NFS V4 host authentication:

NFS servers always identify client hosts by IP addresses and host names, regardless of the authentication method that you use. When Kerberos authentication is the only allowed security method for an exported directory, the NFS client session must be properly authenticated before gaining access to any of the data in that directory.

NFS V4 normally authenticates clients at the user level rather than at the host level. The two user authentication methods are **auth_sys** (UNIX authentication) and **RPCSEC_GSS** (Kerberos). Under the **auth_sys** security method, the user is authenticated at the client, usually through a logon name and password. The NFS server trusts the user and group identities presented by its clients. When an NFS client and server are using Kerberos 5 authentication, the client and server must establish a security context for NFS requests. The security context is a data structure that indicates that the client and server have completed a mutual authentication procedure. If requested, the context also contains the encryption keys that are used for protecting exchanged data. The security context has a lifetime and might need to be refreshed by the client.

For more information about the **RPCSEC_GSS** authentication process, see the readme files, Network File System security.

NFS V4 host authorization:

Host authorization in an Network File System (NFS) context means controlling which NFS client hosts can mount exported directories from the NFS server. This is accomplished in AIX with a combination of the `/etc/exports` file and the **exportfs** command.

NFS V4 has the security-related options as shown in the following table.

Option	Description
vers	Controls which version NFS mounts you can use. Possible values are 2, 3, and 4. Versions 2 and 3 cannot be enforced separately. Specifying Version 2 or 3 allows access by clients using either NFS protocol Versions 2 or 3. Version 4 can be specified independently and must be specified to allow access by clients using Version 4 protocol. The default value in NIM is 3. Valid values in NIM are 3 and 4.

Option	Description
sec	Controls which security methods can be used. Possible values are: **sys** UNIX authentication, *default option* **dh** DES authentication **krb5** Kerberos, authentication only **krb5i** Kerberos, authentication, and integrity **krb5p** Kerberos, authentication, integrity, and privacy **none** Allows mount requests to proceed with anonymous credentials The default value in NIM is **sys**. Valid values in NIM are **sys** and **krb5**.

The **sec** option can appear more than once in the exports definition for a directory. This allows different access options, such as **ro**, **rw**, and **root**, to be specified for the different security options. For example, hosts using the **sys** security method might only be allowed read access, while hosts using the **krb5** security method might be allowed read and write access.

Using NIM in NFS, there is a standard set of export options that you can use. You can also use user-defined options, but they require you to manage NFS exports for the directory or file system using the NFS export commands, such as **mknfsexp**, **chnfsexp**, and **rmnfsexp**. These export options are separate from NIM export options.

Prerequisites for setting up a NIM environment with NFS security using Kerberos 5:

Your system must meet these prerequisites before you can configure Kerberos 5.

- The NIM master must have AIX Version 7.1 or later installed.
- The NIM master must be configured.
- IBM Network Access Server (NAS) Version 1.4 or later from the *AIX Expansion Pack CD* server files must be installed:
  - `krb5.lic`
  - `krb5.client`
  - `krb5.server`
  - `modcrypt.base`
- Kerberos services must be configured and authenticated with the Key Distribution Center (KDC) server.
- Any participating NIM clients must have AIX 6.1 or later installed.
- IBM NAS Version 1.4 or later from the *AIX Expansion Pack CD* client files must be installed:
  - `krb5.lic`
  - `krb5.client`
  - `modcrypt.base`
- The Kerberos client must be configured and authenticated with the KDC server.

While NIM is capable of configuring NFS V4, due to the variation of Kerberos configurations, you must manage the KDC configuration and services outside of NIM. Use the **sec** option in the NIM database for export-list generation only. You can use the sample scripts in the `bos.sysmgt.nim.client` fileset to set up Kerberos. After Kerberos 5 is configured in the NIM environment, you must authenticate and obtain tickets for each client and the NIM master. Use the **usr/krb5/bin/kinit** command for ticket-granting options.

For additional help for NIM and Kerberos 5, see the `/usr/lpp/bos.sysmgt/nim/README` file.

Managing NFS client communication options using SMIT:

Use the following procedure to configure the NFS client communication options using SMIT.

- Type `smitty nim_global_nfs` on the NIM client.
- Select any of the NFS client options as shown in the following table.

Option	Value
Enable/Disable Global Usage of NFS Reserved Ports?	Specifies that a non-reserved IP port number is to be used. The value is `disable`. A value of `enable` uses a reserved IP port number when the NFS client-communicates with the NFS server.
Allow NIM to enable port-checking on NIM master?	Checks whether an NFS request originated from a privileged port. The default value is `no`. A value of `yes` directs the NFS server to do port checking on the incoming NFS requests.
Specify the NFS Local Domain	Specify that the NFS local domain of the system should be changed. The value that you specify is used to create the NIM environment attribute **nfs_domain** and is used as the domain name in the `/etc/nfs/local_domain` file. **Note:** You must set this option before exporting NIM resources as NFS V4 mounts.

Managing NFS client-communication options from the command line:

Use these commands to configure NFS client-communication options on the NIM master.

Use the following commands on the NIM master:

- To enable global usage of NFS reserved ports, use the following command:
  ```
  nim -o change -a nfs_reserved_port=yes master
  ```
- To disable global usage of NFS reserved ports, use the following command:
  ```
  nim -o change -a nfs_reserved_port=no master
  ```
- To enable port checking on the NIM master NFS server, use the following command:
  ```
  nfso -o portcheck=1
  ```
- To disable port checking on the NIM master NFS server, use the following command:
  ```
  nfso -o portcheck=0
  ```
- To create a simple KDC server and principals on the NIM master NFS server, use the following command:
  ```
  /usr/samples/nim/krb5/config_rpcsec_server -p <password> -u <user principal name>
  ```
  This command creates a new-system user name based on the principal name and password provided. See Sample KDC Server Definition File.
- To delete the KDC server and principals on the NIM master NFS server, use the following command:
  ```
  /usr/sbin/unconfig.krb
  ```
  This command removes all Kerberos 5 configuration information.

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 141.

Listing software installed on a standalone client or SPOT:

You can list software installed on a standalone client or SPOT using SMIT, or the command line.

Listing software installed on a standalone client or SPOT using SMIT:

Follow this procedure for listing software installed on a standalone client or SPOT using SMIT.

1. 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.

Listing software installed on a standalone client or SPOT from the command line:

Follow this procedure for listing software installed on a standalone client or SPOT 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:

You can list software updates, installed on a standalone client or SPOT, by keyword using the SMIT, or the command line.

Listing software updates, installed on a standalone client or SPOT, by keyword using SMIT:

Follow this procedure for listing software updates, installed on a standalone client or SPOT, by keyword using 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.

Listing software updates, installed on a standalone client or SPOT, by keyword from the command line:

Follow this procedure for listing software updates, installed on a standalone client or SPOT, by keyword from the command line.

Enter the following command:

```
nim -o fix_query [ -afixes="FixKeywords" ] \
[-afix_bundle=FixBundleName ] [ -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:

Item	Description
-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:

This kind of task is accomplished by performing the NIM **maint** operation on a **SPOT** using NIM application, SMIT, or command line interface.

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 **SPOT** resources as well. For example, you can remove previously installed optional software from a **SPOT** when they are no longer being used. You interact with the **installp** command by supplying the **installp_flags,** and either **filesets** or **installp_bundle** attributes.

Maintaining software on standalone clients and SPOT resources using SMIT:

Follow this procedure for software maintenance on standalone clients and SPOT resources using SMIT.

1. 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.

Maintaining software on standalone clients and SPOT resources from the command line:

Follow this procedure for maintaining software on standalone clients and SPOT resources 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` software package from `standalone1`, enter:

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

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

  ```
  nim -o maint -F -a filesets=bos.adt -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
  ```

Maintaining software in an lpp_source

To add or remove software in an **lpp_source**, 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:

You can copy software to an lpp_source using the SMIT, or the command line.

Copying software to an lpp_source using SMIT:

Follow this procedure for copying software to an lpp_source using 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.

Copying software to an lpp_source from the command line:

Follow this procedure for copying software to an lpp_source 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.

Removing software from an lpp_source from the command line:

Follow this procedure for removing software from an lpp_source 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. You can run the NIM check operation from SMIT, or the command line.

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 the **lpp_source** contains the images necessary to install the Base Operating System images on a machine.

Running the NIM check operation using SMIT:

Follow this procedure for running the NIM check operation using SMIT.

1. 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.

Running the NIM check operation from the command line:

Follow this procedure for running the NIM check operation 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
```

Managing the NIM master

Tasks for managing the NIM master are described.

For additional information on NFS V4, see Network File System in *Networks and communication management*.

Deactivating the NIM master and removing the NIM master fileset:

After 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 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 can NFS-export a resource:

Follow these instructions to increase the number of hosts to which NIM can NFS-export a resource.

By default, when NIM exports a file or directory through 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 approximately one-half, effectively permitting files to be allocated to a greater number of hosts. This action 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 198.

Controlling the asynchronous behavior of NIM operations:

Certain NIM operations are asynchronous, meaning that NIM master might initiate the operation on the client, but does not wait for the operation to finish. The reason for this asynchronous behavior is because the NIM operation running 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 the **cust**, **maint**, and **lppchk** operations 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:

Follow these instructions to suppress 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 indicate to NIM not to display output from the **installp** 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. You can reduce space consumption significantly on resource servers by creating **/usr SPOTs** and defining CD-ROM file-system directories as **lpp_sources**.

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 BOS 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 "Using the SPOT resource" on page 253 and "Defining /usr versus non-/usr SPOTs" on page 150.

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. For more information about using a CD-ROM file system as an **lpp_source**, see "Defining an lpp_source on DVD-ROM versus hard disk" on page 151.

Obtaining support for multiple mksysb operations in NIM:

During the allocation of NIM **mksysb** images, only the file is exported to the NFS clients. However, during **mksysb** creation, the parent directory is also exported. If you use that filesystem to create a **mksysb** image of a system while another system is restoring a **mksysb** image from that filesystem, you will get NFS errors.

To avoid this problem, use the environment variable *NIM_MKSYSB_SUBDIRS* on the NIM master. When this variable is set to `yes`, subdirectories are used to separate **mksysb** images. The subdirectories are transparent to the user, but they provide separate child locations for NFS exporting.

Saving system backup information:

When you are defining a **mksysb** NIM resource using the **nim -o define -t mksysb** command, the **-a mksysb_flags=***xxx* attribute can be given the **-p** option, which prevents the `mksysb` image from being compressed.

Note: Due to the amount of space that this system backup is likely to occupy, the location into which the `mksysb` file is saved, specified by the **-a location=***xxx* attribute, must be large-file enabled. Otherwise, errors can occur.

Managing 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.

A NIM administrator would not want this CPU ID validation to be performed in the following situations:

- 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.
- When a client machine is migrated with Logical Partition Mobility (LPM), giving the client new hardware and a new CPU ID.

Managing client CPU ID validation using SMIT:

Use this information to enable or disable client CPU ID validation from the SMIT interface.

Type the SMIT fast path:

```
smit nim_cpuid_validate
```

Managing client CPU ID validation 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.

Installing and managing software with detached WPARs:

If a system has detached WPARs (WPARs which have separately installed writable `/usr` files), the system software must remain compatible between the global environment and the detached WPARs.

You can use the **inuwpar** command to perform an installation in a global environment. After that installation is complete, the installation continues in all of the detached system WPARs or a specified set of detached system WPARs. If the **-G** flag is specified, the installation is first attempted in the global environment. If the installation succeeds, the installation is attempted in sequence on each of the specified WPARs.

For the **inuwpar** command to be successful, any installation device used for the command must be available in the WPAR. For the best results, ensure that the installation device is in a directory on a local file system in the global environment. The **inuwpar** command attempts to mount the installation device into the WPAR file systems and repeat the operation. If the installation device cannot be mounted, the operation on that WPAR fails, and the installation proceeds to the next WPAR. Block and character devices cannot be used as the installation device for **inuwpar** operations. If the file system is remote, it must be accessible to each WPAR.

If the installation directory is already accessible to the WPARs, the following options prevent the **inuwpar** command from attempting to remount the device into the WPAR:

- Use the **-d** flag to specify the installation directory.
- If the installation device has the same path within the WPAR as in the global environment, us the **-D** flag.

For more information about managing software with detached WPARs, see Managing software with detached workload partitions.

For information about recovering incompatible detached WPARs, see Recovering incompatible detached workload partitions.

Using NIM resources

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 the resources must reside locally on the servers providing these resources, on a JFS or JFS2 file system, because NFS can only export file system objects that are stored on local media in the machines from which they are exported. A large number of resources (files and directories) are needed to support NIM software installation and maintenance operations.

To obtain detailed information about any resource, enter the following from the NIM master:

```
lsnim -Pa ResourceType
```

The SMIT interfaces are designed to hide much of the detail required for the command line interface. Therefore, these sections only document 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.

Using the file_res resource

The **file_res** resource represents a directory where network installation management (NIM) allows files to be stored on the server.

When the **file_res** resource is allocated to a client, a copy of the contents of the directory is added to the client at the location that is specified in the **dest_dir** attribute.

Defining a file_res resource:

You can define a **file_res** resource by using the command syntax and attributes.

A **file_res** resource is where NIM allows for resource files to be stored on the server. When the resource is allocated to a client, a copy of the directory contents is placed on the client at a location that is specified by the `dest_dir` attribute.

The command syntax for defining a **file_res** resource follows:

```
nim -o define -t file_res -a Attribute=Value ... file_resName
```

The following attributes are required for the **file_res** resource:

Table 15. Required file_res resource attributes

Item	Description
-a location=*Value*	Specifies the full path name of the directory on the NIM server. This path is used as a source directory among clients.
-a dest_dir=*Value*	Specifies the full path name of the directory on the NIM client. This path is where the source directory is recursively copied into. **Notes:** • If the target directory does not exist on the destination machine, the entire source directory contents are copied (including the hidden files in the top-level directory). • If the target directory exists on the destination machine, the source directory contents are copied (excluding the hidden files in the top-level directory).
-a server=*Value*	Specifies the name of the machine where the directory for the **file_res** resource is created.

The following attributes are optional for the **file_res** resource:

Table 16. Optional file_res resource attributes

Item	Description
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which the file_res resource must be added.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a nfs_vers=*Value*	Specifies the Network File System (NFS) protocol version that is required for NFS access.
-a nfs_sec=*Value*	Specifies the security method that is required for NFS access.

Using NIM groups

NIM groups are essentially lists of machines and resources. Groups can be used in NIM operations to simplify repetitive administrative tasks.

Working with NIM 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) and of the same architecture, because some NIM operations are restricted to certain target types.

The first member added to a group determines the architecture and 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 group members from operations on the group" on page 212.

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 212 and "Removing members from machine groups" on page 197.

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:

Item	Description
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:

Item	Description
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. For information on how to increase the number of machines to which a resource can be allocated, refer to "Exporting NIM resources globally" on page 198.

Working with NIM 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:

Item	Description
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
```

NIM task road map

The following are NIM configuration tasks and installation tasks and where they can be found in this topic. Also provided is a brief description of the task. Where appropriate, the SMIT fast path is provided.

Table 17. NIM task road map

Item	Description	
NIM Task	**SMIT fast path**	**Description**
"Using EZNIM" on page 282	smit eznim	Configure the NIM enviroment using EZNIM. Allows you to configure your system as a NIM master or a NIM client. If you configure your system as a NIM master, EZNIM also creates the minimum basic installation resources.
"Configuring the NIM master and creating basic installation resources" on page 121	smit nim_config_env	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.
"Adding standalone clients to the NIM environment" on page 134	smit nim_mkmac	Describes how to add standalone clients to the NIM environment.
"Using installation images to install the base operating system on a NIM client" on page 167	smit nim_bosinst	Describes how to perform a BOS installation on a NIM client.
"Using a mksysb image to install the base operating system on a NIM client" on page 169	smit nim_bosinst	Describes how to restore a **mksysb** image and additional software to a target NIM client from a **mksysb** resource in the NIM environment.
"Performing a nonprompted BOS installation" on page 168	• smit nim_mkres • smit nim_bosinst	Provides information about how to perform a nonprompted NIM BOS installation using a **bosinst_data** resource.
"Using NIM with ATM networks" on page 139		Provides information about how to configure NIM to work with ATM adapters.
"Using installation images to install the base operating system on a NIM client" on page 167	smit nim_task_inst	Describes how to use NIM to install software packages, updates, and maintenance levels on running, configured NIM clients and **SPOT** resources.
"Performing boot diagnostics on NIM clients" on page 192	smit nim_mac_op	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. 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.
"Maintaining software on standalone clients and SPOT resources" on page 219	smit nim_task_maint	Provides information about how to commit, reject, remove, copy, verify, and clean up software.

Table 17. NIM task road map (continued)

Item	Description	
"Adding a diskless or dataless client to the NIM environment" on page 147	smit nim_task_dd	Provides information about how to add diskless and dataless systems to your NIM environment. You can also manage resources for diskless and dataless clients from the NIM master.
"Installing to an alternate disk on a NIM client" on page 174	smit nim_alt_install	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.
"Performing an alternate disk migration installation" on page 70	smit nimadm	NIM can be used to perform an alternate disk migration installation to a NIM client.

Using the adapter_def resource

The **adapter_def** resource represents a directory that contains secondary adapter configuration files that are used during **bos_inst** and **cust** operations.

The **adapter_def** resource directory is populated with secondary-adapter configuration files by the **nimadapters** command.

Defining an adapter_def resource:

You can use the following syntax and attributes for defining an **adapter_def** resource.

The command line syntax for defining an **adapter_def** resource is:

```
nim -o define -t adapter_def -a Attribute=Value ... adapter_defName
```

The following attributes are required for the **adapter_def** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the **adapter_def** resource directory.
-a server=*Value*	Specifies the name of the machine where the **adapter_def** resource directory resides. Only the master can serve an **adapter_def** resource.

The following attributes are optional for the **adapter_def** resource:

Item	Description
-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 nfs_vers=*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec=*Value*	Specifies the security method required for NFS access.

Secondary adapter support is available for AIX. Before you enable a secondary adapter, you must verify the AIX version the client is on. The secondary adapters will fail to configure because NIM is unable to find the `/usr/lpp/bos.sysmgt/nim/methods/c_cfgadptrs` client method. The following example shows the outcome if you attempt to enable this support on your NIM master.

```
nim -o cust -a adapter_def=adapter_def1 rspc10
trigger.austin.xyz.com. 0042-001 nim: processing error encountered on "master":
0042-001 m_cust: processing error encountered on "rspc10":
0042-175 c_script: An unexpected result was returned by the
"trigger.austin.xyz.com:/export/nim/scripts/rspc10.script" command:
/tmp/_nim_dir_4714/script[10]: /usr/lpp/bos.sysmgt/nim/methods/c_cfgadptrs:  not found.
```

Distributing 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.

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.

Creating resources in the /tmp directory or /tmp subdirectories

NIM resources should not be created in the `/tmp` directory or `/tmp` subdirectories (including filesystems mounted under `/tmp`).

Creating NIM resources on an NFS shared NAS device

You can use a network-attached storage (NAS) device to store your Network Installation Management (NIM) resources by using the nas_filer resource server.

NIM support allows the hosting of file-type resources (such as mksysb, savevg, resolv_conf, bosinst_data, and script) on a NAS device. The resources can be defined in the NIM server database, and can be used for installation without changing any network information or configuration definitions on the Shared Product Option Tree (SPOT) server.

The `nas_filer` resource server is available in the NIM environment, and requires an interface attribute and a password file. You must manually define export rules and perform storage and disk management before you use any NIM operations.

To create resources on a NAS device by using the `nas_filer` resource server, complete the following steps:

1. Define the `nas_filer` object. You can enter a command similar to the following example:

   ```
   # nim -o define -t nas_filer -a if1="find_net als046245.server.com 0" -a
   passwd_file=/export/nim/pswfile netapp1
   ```

2. Define a `mksysb` file that exists on the NAS device as a NIM resource. You can enter a command similar to the following example:

   ```
   # nim -o define -t mksysb -a server=netapp1 -a location=/vol/vol0/nim_lun1/client1.nas_filer
   NetApp_bkup1
   ```

3. Optional: If necessary, create a new resource (client backup) on the NAS device. You can use the following command to create a `mksysb` resource:

```
# nim -o define -t mksysb -a server=netapp1 -a location=/vol/vol10/nim_lun1/mordor05_bkup -a
source=mordor05 -a mk_image=yes NetApp_mordor05
```

4. Optional: If necessary, copy an existing NIM resource to the `nas_filer` object. You can use the following command to copy a `mksysb` resource.

```
# nim -o define -t mksysb -a server=netapp1 -a location=/vol/vol10/nim_lun1/replicate_bkup -a
source=master_backup NetApp_master_backup
```

Associating and defining NIM resource groups

NIM resource groups allow association and definition 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.

Defining a resource group:

You can use the following procedures to define a resource group.

Defining a resource group using SMIT:

Follow this procedure to define a resource group using 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.

Defining a resource group from the command line:

Follow this procedure for defining a resource group 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:

Use the following procedures to allocate resource groups.

Allocating a resource group using SMIT:

Follow this procedure to allocate a resource group using 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.

Allocating a resource group from the command line:

Follow this procedure to allocate a resource group 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.

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
```

Defining default resource groups using SMIT:

Follow this procedure for defining default resource groups using SMIT.

1. 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.

Defining default resource groups from the command line:

Follow this procedure to define default resource groups from the command line.

Enter:

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

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

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

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.

Restricting NIM client resource allocation using SMIT:

Use this procedure to change NIM client-allocation restrictions from the SMIT interface.

Type the SMIT fast path:

```
smit nim_control_alloc
```

Restricting NIM client resource allocation from the command line:

Use these procedures to restrict NIM client resource allocation 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
```

Using the 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 automatically be deallocated when the operation finishes.

Using the bosinst_data resource

With a **bosinst_data** resource, data can be specified in a NIM resource prior to the installation.

A **bosinst_data** resource represents a file that contains information for the BOS installation program. Normally, the BOS installation program looks for this information in the `/bosinst.data` file in the BOS installation 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 "Performing a nonprompted BOS installation" on page 168 for a sample `bosinst_data` file.

For instructions on how to create and use a `bosinst_data` file, see "Performing a nonprompted BOS installation" on page 168.

Defining a bosinst_data resource:

You can use the following command-line syntax and attributes for 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 the **bosinst_data** resource:

Item	Description
-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 the **bosinst_data** resource:

Item	Description
-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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using the devexports resource

A **devexports** resource represents a file to use as a master device-exports file when you create a WPAR.

This file must match the **devexports** file format. If a **devexports** resource is not allocated when the WPAR is created, the `/etc/wpars/devexports` file on the managing system is used to describe specific device handling when the WPAR is created.

Defining a devexports resource:

You can use the following command-line syntax and attributes for defining a **devexports** resource.

The command line syntax for defining a **devexports** resource is as follows:

```
nim -o define -t devexports -a server=server_name \
-a location=devexports_file_location devexports_object_name
```

After the **devexports** resource is defined, you can use the **devexports** resource to allocate the resource and create a WPAR, as follows:

```
nim -o create -a devexports=devexports_object_name client_name
```

The following attributes are required for the **devexports** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the file being defined as the **devexports** resource.
-a server=*Value*	Specifies the name of the machine where the file for the **devexports** resource resides.

The following attributes are optional for the **devexports** resource:

Item	Description
-a comments=*Value*	Describes the resource.
-a source=*Value*	Specifies an existing **devexports** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using the dump resource

A **dump** resource represents a directory in which client dump directories are maintained.

The dump resource can only be used by a dataless client or a diskless client. The dump resource requires the iSCSI S/W Target package to be installed on the dump resource server. Only POWER6 and later clients that have the appropriate firmware can dump to the dump resource.

When a **dump** resource is allocated to a client, NIM creates a subdirectory identified by the client's name for the client's exclusive use. After initialization, the client uses this directory to store any **dump** images it creates. Note that such dumps are firmware-assisted.

Note: If you subsequently deallocate this resource, NIM removes the `dump` directory and the subdirectory that NIM created for the client's use.

Defining a dump resource:

You can use the following command-line syntax and attributes for 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 the **dump** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the parent directory for the client **dump** directories.
-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 the **dump** resource:

Item	Description
-a dumpsize=*Value*	Specifies the maximum size of a dump, in GB. The minimum value is 2 GB, and the default value is 50 GB. Space is not allocated until a client starts to dump. The dump resource should be large enough to accept and hold the expected number of dump images for the installation. The dump resource is used to store snap data from a snap operation.
-a max_dumps=*Value*	Specifies the maximum number of dumps collected for a client. The default is 1. When a new dump is written to the dump resource, the oldest dump is deleted if the new dump exceeds the maximum number of dumps.
-a notify=*Value*	Specifies the path to an administrator notify method that is invoked when a new dump is captured, or when a dump error occurs on the client.
-a snapcollect=*Value*	Specifies if a snap record must be collected from the failed client after a dump completion. Valid values are yes and no. The default value is no.
-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**.

Using the exclude_files resource

This resource may be used when a **mksysb** resource is being created from a running NIM client.

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.

Defining an exclude_files resource:

You can use the following command-line syntax and attributes for 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 the **exclude_files** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the file containing the list of files and directories to exclude from the **mksysb**.
-a server=*Value*	Specifies the name of the machine where the file for the **exclude_files** resource resides.

The following attributes are optional for the **exclude_files** resource:

Item	Description
-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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using the fb_script resource

Use an **fb_script** resource to provide device-configuration information.

An **fb_script** resource represents a file that is used to configure devices when a NIM client is booting for the first time after the BOS installation process is completed. During BOS installation, certain customization operations (such as device configuration) cannot be performed because they require certain daemons to be running. However, at this point in the BOS installation process, daemons are not available. As a result, certain devices may not be configured during system reboot, and have to be manually configured after the system has booted.

You can use an **fb_script** resource to provide device-configuration information. The BOS installation process adds the content of the **fb_script** resource to the `/etc/firstboot` file, which is run the first time that a client is booted. The `/etc/firstboot` file then performs the device configuration.

For example, you can enable a script to add a message to the message of the day file by completing the following steps when you are using NIM:

1. Create the **fb_script** resource named `myscript.sh` in the `/export/nim/script_res/` directory.

   An example of the content of a script follows:

   ```
   #!/usr/bin/ksh
   echo "Be sure to follow all security guidelines." >> /etc/motd
   ```

2. Create the NIM resource to represent the **fb_script** resource by entering the following command:

```
nim -o define -t fb_script -a server=master -a
location=/export/nim/script_res/myscript.sh fb_script1
```

3. Specify the script that will run during the next installation by entering the following command:

```
nim -o bos_inst -a spot=spot1 -a lpp_source=lpp_source1 -a
 fb_script=fb_script1 -a accept_licenses=yes machA
```

The contents of the script will be appended to the `/etc/firstboot` file and run during the next restart. This resource can be used to control tunable parameters on your system.

Defining an fb_script resource:

You can use the following command-line syntax and attributes for defining a **fb_script** resource.

The command line syntax for defining an **fb_script** resource is as follows:

```
nim -o define -t fb_script -a server=server_name \
 -a location=fbscript_file_location fbscript_object_name
```

After the **fb_script** resource is defined, you can allocate the resource and initiate a BOS installation operation using the **fb_script** resource, as follows:

```
nim -o bos_inst -a fb_script=fbscript_object_name client_name
```

The following attributes are required for the **fb_script** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the file being defined as the **fb_script** resource.
-a server=*Value*	Specifies the name of the machine where the file for the **fb_script** resource resides.

The following attributes are optional for the **fb_script** resource:

Item	Description
-a comments=*Value*	Describes the resource.
-a source=*Value*	Specifies an existing **fb_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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using a 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:

You can use the following command-line syntax and attributes for 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 the **fix_bundle** resource:

Item	Description
-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 the **fix_bundle** resource:

Item	Description
-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 **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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using a home resource

A **home** resource represents a directory in which client `/home` directories are maintained.

When **home** 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:

You can use the following command-line syntax and attributes for 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 the **home** resource:

Item	Description
-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 the **home** resource:

Item	Description
-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**.

Using an image_data resource

An **image_data** resource represents a file that contains information for the BOS installation program. This information describes how physical disks and file systems should be configured in the root volume group during installation.

Normally, the BOS installation program determines default values that should be used, or uses an **image.data** file from a **mksysb** being restored. Use a customized **image_data** resource only in special cases.

A sample **image.data** file (*SPOT_Offset***/usr/lpp/bosinst/image.template**) is located on the **SPOT** resource server. For more information about the `image.data` file, see the *Files Reference*.

Defining an image_data resource:

You can use the following command-line syntax and attributes for 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 the **image.data** resource:

Item	Description
-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 the **image.data** resource:

Item	Description
-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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using an 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:

You can use the following command-line syntax and attributes for 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 the **installp_bundle** resource:

Item	Description
-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 the **installp_bundle** resource:

Item	Description
-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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using an ios_mksysb resource

An **ios_mksysb** resource represents a file that is a system backup image created by using the **backupios** command from a Virtual I/O Server (VIOS). The **ios_mksysb** resource can be used as the source for the installation of VIOS or the installation of an Integrated Virtualization Management (IVM) client management system.

The **ios_mksysb** image must reside on the hard disk of a system in the Network Installation Management (NIM) environment to be defined as a resource. The **ios_mksysb** image cannot reside on a tape or other external media. The VIOS media contains a system backup file. This file must be copied from the media onto the hard disk of a system in the NIM environment. The **ios_mksysb** image on the VIOS media is split into multiple files due to the file size constraint of the physical VIOS media. The split **mksysb** files must be joined together when copied from media to the hard disk of the system. The following example shows the VIOS media mounted in the **/mnt** directory.

```
cat /mnt/nimol/ioserver_res/mksysb \
/mnt/nimol/ioserver_res/mksysb2 > /export/mksysb/vio_mksysb
```

The **mksysb** image can be split across multiple VIOS media. In which case, the **mksysb** image must be joined into a single **mksysb** file by using the **cat** command that is described in the previous example from a multivolume VIOS media.

An **ios_mksysb** resource can be defined from an image that already exists on the hard disk of the NIM master or on any NIM client. If such an image does not exist, the image 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 is the source for the backup, and set the **mk_image** attribute to `yes` in the command to define the **ios_mksysb** resource. To exclude the `/var/vio/VMLibrary` file from the VIOS backup image, use the *backupios_flags* attribute set to the *-nomedialib* value. See the help information for the **backupios** command on the VIOS for acceptable flags when using the **ios_mksysb** command.

Defining an ios_mksysb resource:

You can use the following command-line syntax, flags, and attributes for defining an **ios_mksysb** resource.

The syntax for defining an **ios_mksysb** resource is:

```
nim -o define -t ios_mksysb -a Attribute=Value ... ios_mksysbName
```

The following values are valid for the *backupios_flags* attribute when defining the **ios_mksysb** resource:

- **-nomedialib**
- **-nosvg**

Multiple values can be enclosed within double quotation marks and separated by a space. The following example command would be entered on one line:

```
nim -o define -t ios_mksysb -a location=/nim/mybackup -a server=master \
 -a backupios_flags="-nosvg -nomedialib" <ios_mksysb_object_name>
```

The descriptions of these and other flags are discussed in the *backupios* command information

The following attributes are required for the **ios_mksysb** resource:

Attribute	Description
-a location=Value	Specifies the full path name of the **ios_mksysb** image.
-a server=Value	Specifies the name of the system where the **ios_mksysb** image resides or is to be created.

The following attributes are optional for the **ios_mksysb** resource:

Attribute	Description
-a comments=Value	Specifies the **ios_mksysb** image.
-a mk_image=Value	Specifies the flag to use to create an **ios_mksysb** image from a system in the NIM environment.
-a backupios_flags=Value	Specifies the flags to use when the command creates the backup.
-a size_preview=Value	Specifies the flags to verify that space is available before creating an **ios_mksysb** image.
-a source=Value	Specifies the name of the VIOS or IVM NIM client system to be backed up in the **ios_mksysb** image.
-a verbose=Value	Displays debugging information. To show maximum detail, specify a value of 5.
-a source=Value	Specifies the name of the VIOS machine to be backed up in the **ios_mksysb** image if the *mk_image* attribute is specified. If the *mk_image* attribute is not specified, this value specifies an existing **ios_mksysb** resource to be replicated when you define a new **ios_mksysb** resource. The file pointed to by the source resource is copied to the new location.
-a nfs_vers=Value	Specifies the Network File System (NFS) protocol version required for NFS access.
-a nfs_sec=Value	Specifies the security method required for NFS access.

Related information:

backupios command

Using an 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**.

The recommended types of NIM **lpp_source** resources are as follows:

- A complete (`simages=yes`) NIM **lpp_source** resource that contains AIX base-installation file sets (generated from AIX base-installation media) that are capable of installing the AIX operating system on an AIX machine.
- An update **lpp_source** resource that contains only technology-level or service-pack updates (such as those on AIX update media or available for downloading from an IBM service site).
- An application **lpp_source** resource that contains additional application software (to be installed after a base operating system installation).

If you combine these images into one **lpp_source** resource, use the **lppmgr** command to remove superseded or duplicate images.

Note: Do not combine an **lpp_source** resource that was created from a technology-level base media with images from the same technology-level update media.

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 are:

POWER processor-based	Itanium processor-based
bos	**bos**
bos.64bit	N/A
bos.rte.up	N/A
bos.rte.mp	N/A
bos.up	N/A
bos.mp	**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
```

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 either the name of the device, absolute directory pathname or absolute ISO image pathname that contains the installation images. NIM copies the software images from this source 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.
- If an **lpp_source** should be created from a source device using a list of software based on the **source's simages** list, specify the **use_source_simages** attribute with a value **yes** when defining the **lpp_source**. When using this attribute, you must specify a source attribute and the **packages** attribute cannot be supplied.
- If a warning message of 0042-256 is displayed when you create an **lpp_source** resource, the listed file sets are no longer available for this source level. This issue is a known limitation for prior releases.

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 100 MB, but a default **lpp_source** created from a CD-ROM may be over 350 MB. 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.

The **simages** message displays only if the user is creating an **lpp_source** resource with the default installation packages. The **simages** message will display if the **simages** attribute could not be set for the **lpp_source**. If a user creates an **lpp_source** and specifies a list of packages, the **simages** message is not printed. The **simages** attribute is set correctly, whether or not a **simages** message is printed or not.

If a user attempts to do an **rte** BOS installation with an **lpp_source** that does not have the **simages** attribute, the user receives an error. The error message instructs the user to run **nim -o check** on the **lpp_source** to determine the missing packages needed for an **rte** BOS installation. Whenever a user runs **nim -o check** on an **lpp_source** after it has been created, the **simages** message is printed if the **lpp_source** does not contain all the images needed for a **rte** BOS installation.

Defining an lpp_source resource:

You can use the following command-line syntax and attributes for defining a **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 the **lpp_source** resource:

Item	Description
-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 the **lpp_source** resource:

Item	Description
-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 multi_volume=*Value*	Specifies whether the user should be prompted to insert a second CD into the CD-ROM drive when creating an **lpp_source** with **/dev/cd*** as its source. This attributes's default value is **no**. If the attribute is set to yes, and the server of the resource is not the master, a warning is displayed, and a single-volume **lpp_source** is created.
-a packages=*Value*	Specifies a list of file sets to copy into the **lpp_source** if the default list of images is not desired.
-a show_progress=*Value*	Enables display of informational output when an **lpp_source** is created. The default value for this attribute is **yes**. If the **show_progess** attribute is set to **yes**, and the **server** attribute is set to another machine with an earlier version of the `bos.sysmgt.nim.client` fileset, a warning is displayed indicating that informational output cannot be enabled.
-a source=*Value*	Identifies the source device for copying installation images when defining the **lpp_source**. The value supplied can be either the name of the device, absolute directory path name or absolute ISO image path name that contains the installation images. This attribute is not required if the location of the **lpp_source** already contains the installation images.
-a use_source_simages=*Value*	Specifies whether NIM uses the **simages** package list from what is provided as the **source** attribute. When set to **yes**, NIM uses the **simages** package list from the source specified in the **source** attribute. When set to any other value, NIM defaults to using the NIM master's **simages** package list. A **source** attribute must be provided along with this attribute, and the **packages** attribute cannot be used.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of **5**.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

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.

Item	Description
`power`	POWER processor-based architecture (used for platforms of the type `rs6k`, `rspc`, and `chrp`)

Using the live_update_data resource

A `live_update_data` resource represents a file that contains information for the AIX Live Update operation.

A Live Update operation requires a file that contains information about the client partition, such as the logical partition identifier, the mode to run the Live Update operation, the disk information that the client is running on, and so on.

A `live_update_data` resource can be allocated to a standalone machine as part of the **`cust`** operation.

A sample `lvupdate.data` file (*SPOT_Offset* `/var/adm/ras/liveupdate/lvupdate.template`) is located on the shared product object tree (**SPOT**) resource server.

When you run a Live Update operation, the NIM master sends the client's system information to authenticate the NIM client. This information can be encrypted by installing the `openssl.base` fileset, and by running the **nimconfig -c** command on the NIM master and the **nimclient -c** command on the NIM client.

Defining a live_update_data resource:

You can use the following command-line syntax and attributes to define a `live_update_data` resource.

The command line syntax for defining a `live_update_data` resource follows:

```
nim -o define -t live_update_data -a Attribute=Value ... liveupdateName
```

The following attributes are required to define the `live_update_data` resource:

Attribute	Description
-a location=*Value*	Specifies the full path name of the `live_update_data` resource file.
-a server=*Value*	Specifies the name of the machine where the `live_update_data` resource file is located.

The following attributes are optional to define the `live_update_data` resource:

Attribute	Description
-a comments=*Value*	Describes the resource.
-a group=*Value*	Specifies the name of a resource group to which this resource must be added.
-a verbose=*Value*	Displays information for debugging. To display maximum detail, specify a value of 5.
-a source=*Value*	Specifies an existing `live_update_data` resource to be replicated when you define a new resource. The file that is specified in the **source** attribute is copied to the new location.

Using a mksysb resource

The **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 the mksysb resource:

You can use the following command-line syntax, flags, and attributes for defining the **mksysb** resource.

The command line syntax for defining a **mksysb** resource is:

```
nim -o define -t mksysb -a Attribute=Value ... mksysbName
```

The following flags are valid for the **mksysb** resource:

- **-a**
- **-A**
- **-b**
- **-e**
- **-i**

- **-m**
- **-p**
- **-P**
- **-T**
- **-V**
- **-X**
- **-Z**

For descriptions of these flags, see the **mksysb** command.

The following attributes are required for the **mksysb** resource:

Item	Description
-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 the **mksysb** resource:

Item	Description
-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.
-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.
-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. If the **mk_image** attribute is not specified, 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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using the 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 an **lpp_source**, the **nim_script** resource will be placed on the **lpp_source** server.
- For a **cust** operation without an **lpp_source**, the **nim_script** resource will be placed on the script server.
- Otherwise, the **nim_script** resource will be placed on the NIM master.

Using a 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, 32 MB 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:

You can use the following command-line syntax and attributes for 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 the **paging** resource:

Item	Description
-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 the **paging** resource:

Item	Description
-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**.

Using a resolv_conf resource

A **resolv_conf** resource represents a file containing valid **/etc/resolv.conf** entries that 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:

You can use the following command-line syntax and attributes for defining a **resolv_conf** resource.

The command line syntax for defining a **resolv_conf** resource is:

```
nim -o define -t resolv_conf -a Attribute= ... resolv_confName
```

The following attributes are required for the **resolv_conf** resource:

Item	Description
-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 the **resolv_conf** resource:

Item	Description
-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 **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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using a root resource

A **root** resource represents a directory in which client `root` directories are maintained.

When a **root** 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:

You can use the following command-line syntax for defining a **root** resource.

The command line syntax and attributes for defining a **root** resource is:

```
nim -o define -t root -a Attribute=Value ... RootName
```

The following attributes are required for the **root** resource:

Item	Description
-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 the **root** resource:

Item	Description
-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**.

Using a script resource

A **script** resource represents a file that is a user-defined shell script. After it is defined, this type of resource can be used to perform processing on a client as part of a NIM **cust** or **bos_inst** operation.

The **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 in which the scripts will be run is not predictable.

Note: The **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:

You can use the following command-line syntax and attributes for 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 the **script** resource:

Item	Description
-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 the **script** resource:

Item	Description
-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.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using the savewpar resource

A **savewpar** resource represents a file that is a WPAR backup image that is created by using the **savewar** command. The **savewpar** resource can be used as the source for a WPAR installation.

To be defined as a resource, the **savewpar** image must reside on a NIM-environment machine. It cannot be located on external media.

A **savewpar** resource can be defined from an image on the NIM master or a 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, do the following:

- Specify the name of the NIM WPAR client that will be the source for the backup.
- To define the **savewpar** resource, set the mk_image attribute to `yes` in the **savewar** command.
- Use an **exclude_files** resource to list any files and directories that should not be included in the backup image.

Defining a savewpar resource:

You can use the command-line syntax and attributes to define a **savewpar** resource.

The command line syntax for defining a **savewpar** resource is as follows:

```
nim -o define -t savewpar -a server=server_name \
-a location=savewpar_file_location -a source=wpar_name \
-a mk_image=yes savewpar_object_name
```

After the **savewpar** resource is defined, you can use the **savewpar** resource to allocate the resource and create a WPAR, as follows:

```
nim -o define -t savewpar -a Attribute=Value ... savewparName
```

The following flags are valid for the **savewpar** resource: -A, -a, -b, -e, -i, -m, -N, -p, -V, -v, -X, -Z.

For a description of the **savewpar** resource valid flags, see the **savewpar** command.

The following attributes are required for the **savewpar** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the file being defined as the **savewpar** resource.
-a server=*Value*	Specifies the name of the machine where the file for the **savewpar** resource resides or is created.

The following attributes are optional for the **savewpar** resource:

Item	Description
-a comments=*Value*	Describes the resource.
-a exclude_files=*Value*	Specifies an **exclude_files** resource that is used to exclude files and directories from the system backup.
-a group=*Value*	Specifies the name of a resource group to which this resource should be added.
-a mk_image=*Value*	Specifies the flag that is used to create a **savewpar** image from a machine in the NIM environment.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.
-a savewpar_flags=*Value*	Specifies the flags that are used in the command to create the backup.
-a source=*Value*	Specifies the name of the machine to be backed up in the **savewpar** image if the mk_image attribute is specified. If the mk_image attribute is not specified, this value specifies an existing **savewpar** resource to be replicated when defining a new **savewpar** resource. The file pointed to by the source resource will be copied to the new location.
-a verbose=*Value*	Displays information for debugging. To show maximum detail, specify a value of 5.

Using the secattrs resource

A **secattrs** resource represents a file to use as a master-privileges file when creating a WPAR.

If a **secattrs** resource is not allocated when the WPAR is created, the `/etc/wpars/secattrs` file on the managing system is used to assign the initial set of privileges associated with a WPAR when it is created.

Defining a secattrs resource:

You can use the following command-line syntax and attributes for defining a **secattrs** resource.

The command line syntax for defining a **secattrs** resource is as follows:

```
nim -o define -t secattrs -a server=server_name \
-a location=secattrs_file_location secattrs_object_name
```

After the **secattrs** resource is defined, you can use the **secattrs** resource to allocate the resource and create a WPAR, as follows:

```
nim -o create -a secattrs=secattrs_object_name client_name
```

The following attributes are required for the **secattrs** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the file being defined as the **secattrs** resource.
-a server=*Value*	Specifies the name of the machine where the file for the **secattrs** resource resides.

The following attributes are optional for the **secattrs** resource:

Item	Description
-a comments=*Value*	Describes the resource.
-a source=*Value*	Specifies an existing **secattrs** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Using a 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 a **shared_home** 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 changes only the client's configuration, so the client no longer uses this directory. NIM does not remove the common directory.

Defining a shared_home resource:

You can use the following command-line syntax and attributes for 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 the **shared_home** resource:

Item	Description
-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 the **shared_home** resource:

Item	Description
-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**.

Using a shared_root resource

A **shared_root** resource represents a directory that can be used as a **root** (/) directory by one or more diskless clients. The **shared_root** resource offers drastic improvements in deployment times compared to the regular root resource, especially on large machine groups.

Anytime the client performs a network boot, the client STNFS mounts the directory specified as the **shared_root** directory over the **root** (/) directory to gain access to the **shared_root** directory. The **shared_root** directory remains mounted over the **root** (/) directory on the client as long as the client is running.

Because STNFS is used to mount the **shared_root** directory, any change made by a client to its **root** file system is kept local and is invisible to other clients and to the server of the **shared_root** resource. Any change a client makes to its **root** file system is also lost when the client is rebooted.

You can only use a **shared_root** resource with stateless clients. A regular **root** resource is required for clients that need persistence across reboots.

Note: Whenever the **shared_root** resource is deallocated, NIM changes only the client's configuration, so the client no longer uses the **shared_root** directory. NIM does not remove the common directory.

Note: Performing the **sync_roots** operation may leave the **shared_root** resource in the **sync_roots Rstate** which prevents the resource from being used. As a workaround, either redefine the **shared_root** resource or forcefully reset the master object to reset the **shared_root** state.

Defining a shared_root resource:

Use the **nim** command to define a `shared_root` resource.

To define a **shared_root** resource, use the following command-line syntax:

```
nim -o define -t shared_root -a Attribute=Value ... SharedRootName
```

The following parameters are required for the **shared_root** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the directory to use as a common / (root) directory among clients.
-a server=*Value*	Specifies the name of the system where the **shared_root** resource is created.
-a spot=*Value*	Specifies the name of the **SPOT** resource used to create the **shared_root** resource.

The following parameters are optional for the **shared_root** resource:

Item	Description
-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.

The following example defines a **shared_root** named `my_local_shroot` based on the **SPOT** resource named `my_local_spot`. The **shared_root** directory `/export/my_local_shroot` will be located on the master because the `-a server=master` parameter is specified. This command must be run on the master, and `my_local_spot` must be located on the master. The **SPOT** resource and the **shared_root** resource must be located on the same machine.

```
# nim -o define -t shared_root           \
      -a server=master                   \
      -a location=/export/my_local_shroot \
      -a spot=my_local_spot              \
      my_local_shroot
```

The following example defines a **shared_root** resource named `my_remote_shroot` based on the **SPOT** resource `my_remote_spot`. The **shared_root** directory `/export/my_remote_shroot` will be located on NIM client named `my_client` because the `-a server=my_client` parameter is specified. The command must be run on the master, but the `my_remote_spot` resource must be located on the `my_client` NIM client.

```
# nim -o define -t shared_root             \
      -a server=my_client                  \
      -a location=/export/my_remote_shroot \
      -a spot=my_remote_spot               \
      my_remote_shroot
```

Using the SPOT resource

The **Shared product Object Tree (SPOT)** is a fundamental resource in the NIM environment. It is required to install or initialize all types of machine configurations.

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.

You can create a **SPOT** by converting 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 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: Do not create a non-**/usr SPOT** in a subdirectory of the **/usr** file system.

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

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

The Micro Channel-based systems support booting from the network using Token-Ring, Ethernet, or FDDI. The POWER processor-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 "Customizing NIM clients and SPOT resources" on page 141.

Using network boot images for AIX 4.3 or later SPOTs:

You can use network boot images for AIX 4.3 or later to reduce the amount of disk space used and the time required to create boot images from SPOT resources.

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

Note: Due to kernel changes, AIX 5.2 or later, does not provide NIM support to create or use AIX 4.2 or earlier SPOTs.

In AIX 5.3 with 5300-03 or later, to create SPOT resources for AIX 4.3.3, the environment variable INST_DEBUG must be set as shown:

export INST_DEBUG=yes

If the INST_DEBUG variable is not set, NIM cannot create SPOT resources for AIX 4.3.3.

Defining a SPOT resource:

You can use the following command-line syntax and attributes for 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 the **SPOT** resource:

Item	Description
-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 value supplied can be either the name of the **lpp_source**, the name of the device, absolute directory pathname or absolute ISO image pathname that contains the installation images. To define a minimal SPOT for a **mksysb** installation, the source can be a **mksysb** NIM resource. For a **mksysb** installation of a Virtual I/O Server, the source can be an **ios_mksysb** NIM resource.

The following attributes are optional for the **SPOT** resource:

Item	Description
-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**.
-a nfs_vers=*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec=*Value*	Specifies the security method required for NFS access.

Note: The creation of a **SPOT**, by default, produces a large amount of output. Be sure to scan the output to look for nonfatal errors and warnings that may not be evident from a successful return code.

Using the wpar_spec resource

A **wpar_spec** resource represents a specification file that defines the characteristics of a WPAR that is created on a managing system. For a comprehensive description of the format and permitted contents of a specification file, see the `/usr/samples/wpars/sample.spec` file.

If a **wpar_spec** resource is not allocated when the WPAR is created, the flags specified for the operation and the command defaults are used to create the WPAR.

Defining a wpar_spec resource:

You can use the following command-line syntax and attributes for defining a **wpar_spec** resource.

The command line syntax for defining a **wpar_spec** resource is as follows:

```
nim -o define -t wpar_spec -a server=server_name \
-a location=wpar_spec_file_location wpar_spec_object_name
```

After the **wpar_spec** resource is defined, you can use the **wpar_spec** resource to allocate the resource and create a WPAR, as follows:

```
nim -o create -a wpar_spec=wpar_spec_object_name client_name
```

The following attributes are required for the **wpar_spec** resource:

Item	Description
-a location=*Value*	Specifies the full path name of the file being defined as the **wpar_spec** resource.
-a server=*Value*	Specifies the name of the machine where the file for the **wpar_spec** resource resides.

The following attributes are optional for the **wpar_spec** resource:

Item	Description
-a comments=*Value*	Describes the resource.
-a source=*Value*	Specifies an existing **wpar_spec** resource to be replicated when defining a new resource. The file pointed to by the source resource will be copied to the new location.
-a nfs_vers*Value*	Specifies the NFS protocol version required for NFS access.
-a nfs_sec*Value*	Specifies the security method required for NFS access.

Creating a SPOT resource from the mksysb resource

Creating a SPOT from a **mksysb** resource allows you to only support the devices the **mksysb** uses.

Because the SPOT only contains boot images for the devices in the source **mksysb**, it is significantly smaller than a SPOT created from an installation image. A SPOT that you create from the **mksysb** resource is typically 30 to 50 MB, whereas a SPOT that you create from an installation image is typically 300 MB. You also do not need an **lpp_source** resource with this method.

In AIX 5.3, NIM only creates the **mp** boot image because that is all that is required to boot the SPOT. Similarly, in AIX 7.1, and later, NIM only creates the 64-bit **mp** boot image to boot the SPOT.

You should only use a SPOT created from a **mksysb** to perform **bos_inst** operations of the **mksysb**. Performing other operations on standalone clients, such as **maint** and **diag** is not supported. Performing **cust** operations on a SPOT taken from a **mksysb** is also not supported.

The **mksysb_source** attribute is used to distinguish a SPOT created from a **mksysb** from other SPOTs. The **mksysb_source** attribute is set to the resource name of the **mksysb** you used to create the SPOT.

The **mksysb** from which you create a SPOT can be at any level greater than 4.3.3.75.

Creating a SPOT resource from the mksysb resource with SMIT or from a command line:

To create a SPOT from the **mksysb** on the command line, set the source attribute to the name of the **mksysb** resource you want to create the SPOT from.

If the **mksysb** resource is called "mksysb1," you would use the following command to create a SPOT called "spot1" served by the Network Installation Manager (NIM) master:

```
nim -o define -t spot -a source=mksysb1 -a server=master -a location=/export/spot spot1
```

To create a SPOT from a **mksysb** with SMIT, you can use fast paths **nim_mkres_spot_only** and **nim_mkres**. When the **source** attribute is expanded, all available **mksysb** resources, as well as other sources, are displayed as possible sources for the creation of a SPOT.

Limitation: When NIM creates the SPOT from the **mksysb** resource, the SPOT size cannot be calculated because the `image.data` size in the **mksysb** resource does not reflect the files that are being extracted from the **mksysb** resource. NIM cannot determine the accurate size to increase the file system to store the **mksysb** SPOT. Ensure that there is a buffer size of 1-2 GB in the file system to create the **mksysb** SPOT.

Using a 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:

You can use the following command-line syntax and attributes for 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 the **tmp** resource:

Item	Description
-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 the **tmp** resource:

Item	Description
-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**.

Using NIM operations

A large number of operations can be performed to manage a NIM environment and perform software installation and maintenance.

The 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 |TargetNames
```

Item	Description
cust operation	**lslpp** operation
lppchk operation	**showres** operation
sync_roots operation	**maint** operation
fixquery operation	

Using the NIM activate operation

Use the **activate** operation to start a managed system. The client must have a valid mgmt_profile for the managing system, and the managing system must be running.

The command line syntax for the **activate** operation is as follows:

```
nim -o activate -a Attribute=Value ... TargetName|TargetNames
```

The target of an **activate** operation can be a WPAR client or group of WPAR clients.

There are no required attributes for the **activate** operation. The following optional attributes can be specified for the **activate** operation:

-a cmd_flags=Value
Specifies flags to pass through to the operation on the managing system to activate the system. For WPAR clients, see the **startwpar** command for allowable flags.

-a group=Value
Specifies the name of a WPAR group to use for the operation.

-a show_progress=Value
Indicates whether status should be displayed as the operation is performed. The default value is `show_progress=yes`.

Using the NIM allocate operation

Use the **allocate** operation to make resources available to NIM clients for subsequent operations.

The command line syntax for the **allocate** operation is as follows:

```
nim -o allocate -a ResourceType=ResourceName ... TargetName|TargetNames
```

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:

Item	Description
-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.

Using the NIM alt_disk_install operation

You can use the **alt_disk_install** operation to install a **mksysb** image on a client system alternate disk or disks or to clone a client that is running **rootvg** to an alternate disk.

The **alt_disk_install** operation (available in AIX 6.1 TL9 or later) can also be used to copy the current root volume group of a VIOS or IVM NIM object to an alternate disk and in addition to update the operating system to the next fix pack level.

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 |TargetNames
```

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 |TargetNames
```

The target of an **alt_disk_install** operation can be a standalone NIM client or a group of standalone NIM clients. The clients must also 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:

Item	Description
-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.

The following are required attributes for the **alt_disk_install rootvg** clone operation:

Item	Description
-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 shows 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:

Item	Description
-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 force=*Value*	Specifies whether to skip the checks on the **target_disks**.*Value* can be `yes` or `no`, where `no` is the default value. When set to `yes`, the equivalent of the *–g* flag is passed to the **alt_disk_install** command.
-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.

Item	Description
-a phase=*Value*	Specifies the ***phase*** to run during this invocation of **alt_disk_install**. The installation is divided into three phases, and the default is to perform all three phases. The valid values are; 1, 2, 3, 12, 23, or all. • 12 - performs phases 1 and 2. • 23 - performs phases 2 and 3. • all - performs all 3 phases Refer to the **alt_disk_install** Command reference for more details on phase execution and operational behavior.
-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:

Item	Description
-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`.

Related information:

alt_disk_install

Using the NIM alt_disk_install operation to Clone a VIO Server Disk:

You can use the **alt_disk_install** operation (available in AIX® 6.1 TL9 or later) to clone a **VIO** server's running **rootvg** to an alternate disk or disks.

The NIM **alt_disk_install** operation can be used to copy the current root volume group of a **VIOS** or **IVM NIM** object to an alternate disk and in addition to update the operating system to the next fix pack level.

The command-line syntax for the **alt_disk_install** clone operation of a **VIOS** or **IVM** object is as follows:

```
nim -o alt_disk_install -a source=rootvg -a disk=target_disk(s) \
-a attribute=Value.... TargetName
```

The management target of an **alt_disk_install** operation can be a **VIOS** or **IVM** NIM management object. The management object must also 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 the **alt_disk_install rootvg** clone operation:

Item	Description
-a source=rootvg	Specifies the type of **alt_disk_install** to perform.
-a disk=*target_disk*	Specifies the disks on the client system that the **mksysb** image is restored. This disk or these disks must not currently contain any volume group definition. The **lspv** command shows 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 quotation marks; for example, `'hdisk2 hdisk3'`. When you are specifying a target disk or disks, it is advised that the **lsmap** command must be used (on the target VIOS / IVM) to verify that the target disks are not in use. The **lsmap** command displays the mapping between the virtual host adapters and the physical devices they are backed to.

The following are optional attributes that can be specified for the **alt_disk_install rootvg** clone operation when you are using a VIOS or IVM as the target:

Item	Description
-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 skip the checks on the **target_disks**.*Value* can be `yes` or `no`, where `no` is the default value. When set to `yes`, the equivalent of the *–g* flag is passed to the **alt_disk_install** command.
-a exclude_files= *Value*	Specifies an **exclude_files** resource to use to exclude files and directories from the **rootvg**. Files and directories that are specified in this file is not 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_bundles** resource that lists the APARs to install into the alternate **rootvg** after the clone of the running **rootvg**.
-a force= *Value*	Specifies whether to skip the checks on the **target_disks**. *Value* can be `yes` or `no`, where `no` is the default value. When set to `yes`, the equivalent of the *–g* flag is passed to the **alt_disk_install** command.
-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*.
-a phase= *Value*	Specifies the ***phase*** to run during this invocation of **alt_disk_install**. The installation is divided into three phases, and the default is to perform all three phases. The valid values are; 1, 2, 3, 12, 23, or all. • 12 - performs phases 1 and 2. • 23 - performs phases 2 and 3. • all - performs all 3 phases

Item	Description
-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 are copied from the running system to the `/alt_inst` file systems before the reboot. During this operation, files are copied or modified in the alternate file system because the logical volume names are changed to match the **rootvg**, and they are not accessible until the system is rebooted with the new alternate **rootvg**.
-a set_bootlist= *Value*	Specifies whether to set the `bootlist` to point to the new **rootvg** when the installation is complete. *Value* can be `yes` or `no`, where `yes` is the default value. The next time that the system is rebooted, it will boot from the newly installed alternate disk if *Value* is set to `yes`.
-a verbose= *Value*	Specifies whether to show files as they are being backed up for a rootvg clone. *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`.

Related information:

alt_disk_install

Using the NIM bos_inst operation

Use the **bos_inst** operation to install the AIX BOS on standalone clients.

Note: The following operation is not allowed when resources with architectures different from the client are allocated to the client.

The command line syntax for the **bos_inst** operation is as follows:

```
nim -o bos_inst -a source=Value -a Attribute=Value ... TargetName|TargetNames
```

The target of a **bos_inst** operation can be a standalone NIM client or a group of standalone NIM clients.

The following NIM resources 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** resource is only required for an **rte** installation. The **lpp_source** resource specified must have the **simages** attribute set. However, if you are performing a **bos_inst** operation using a **mksysb** resource and an **lpp_source** resource, then the **simages** attribute is optional. 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**.

mksysb

Installs the machine from a **mksysb** image.

spot Installs the machine from a **SPOT** copy.

Note: If a **SPOT** copy is not complete, the installation will succeed, but the target machine might not be bootable. A **SPOT** copy must have the proper device support to boot the target system. While installing from a **SPOT** copy is the fastest installation method, using **rte** or**mksysb** is more reliable and functional.

Note: A **SPOT** copy will also install the file sets that are part of the **BOS.autoi** bundle.

-a spot=*Value*

Identifies the **SPOT** resource to be used. The **SPOT** provides support for network boot and operations in the boot environment.

The following NIM resources 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. Before the installation process can complete, this attribute must be set to **yes**. The default value is **accept_licenses=no**. If the **bosinst_data** resource resides on the NIM master, the **ACCEPT_LICENSES** field in the **bosinst_data** resource can also be set to **yes**. You can also set the **NIM_LICENSE_ACCEPT** global environment variable to `yes` on the NIM master.

-a adapter_def=*Value*
Specifies the directory containing secondary adapter definition files. The **nimadapters** command parses a secondary-adapters stanza file to build the files required to add NIM secondary adapter definitions to the NIM environment as part of the **adapter_def** resource. The **nimadapters** command does not configure secondary adapters. The actual configuration takes place during a **nim -o bos_inst** or **nim -o cust** operation that references the **adapter_def** resource.

-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 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 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 nonprompted 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 "Using the force_push attribute" on page 266 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** resource used for the installation.

If the level of the **SPOT** resource is greater than the level of the **mksysb** resource, then an **lpp_source** resource must be used and match the level of the **SPOT** resource. When this situation occurs, an update operation is performed by default.

-a no_client_boot=*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 physical_loc=*Value*
Specifies the physical location code or AIX location code of the installation disk to the BOS installation process. This attribute allows you to specify the location code for the installation disk or disks on the command line, and allows you to have a *generic* **bosinst.data** file that does not contain location code information.

To determine a disk's physical location code, type the following:

```
lsdev -Cc disk -l hdisk0 -F "name physloc"
```

For more information about location codes, see Device Location Codes in *Operating system and device management* .

-a preserve_res=*Value*
Indicates whether 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 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 **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, the following occurs:

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 installation. 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 installation program.

NIM BOS installation data:

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` filesystem.

In the NIM environment, this image can come from one of the 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

A **spot** and **lpp_source** are always required to support the **bos_inst rte** operation. A **bos_inst mksysb** operation only requires a **spot** resource be used.

To indicate which BOS image to use, 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. 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.

mksysb
Using **mksysb** as the source results in a target machine that has the same configuration as the machine from which the **mksysb** image was created. This may save installation and configuration time. The **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 message is displayed and user intervention is then required at the target to issue the BOOTP request using the IPL ROM.

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 these resources. 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.

Using the 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.

Using the 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.

Using the 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**. 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.

Using the 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 the **preserve_res** is set to **yes**, any resources that reside in file systems, which are being preserved by the BOS installation process, will also be preserved.

accept_licenses attribute:

The **accept licenses** attribute can be used to control when license acceptance takes place.

If **accept_licenses=yes** is specified, license acceptance takes place automatically as packages are installed. If it is set to **no**, the user is prompted at the client to accept software licenses after the client is rebooted. The default is **accept_licenses=no**.

Using the NIM change operation

Use the **change** operation to modify attributes of NIM objects.

The command line syntax is as follows:

```
nim -F -o change -a Attribute=Value ... TargetName|TargetNames
```

Item	Description
-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.

Using the NIM check operation

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 |TargetNames
```

The target of a **check** operation can be any NIM client, a group of NIM clients, a Virtual Input or Server client, a SPOT resource, or a LPP_Source resource.

The flags and attributes that can be specified for the **check** operation are as follows:

Item	Description
-F (optional)	Notifies the 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, the flag forces the SPOT network boot images to be rebuilt. The **-F** flag is not required when you perform the **check** operation on client machines. If the**-F** flag is used in a check operation on a client machine, the **default_profile** attribute is re-created in case the attribute is old.
-a debug=*Value* (optional)	Builds network boot images for aSPOT network 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**. For more information on the **debug** attribute, refer to "Producing debug output from a network boot image" on page 316.

When applied to NIM clients, the **check** operation updates the machine state (**Mstate**) of the client. A ping test is performed to check whether the client is reachable. After the **check** operation is performed, the **Mstate** of the client is set to either **running** or **not running**.

When the **mgmt_profile** attribute is set, the check operation checks the related HMC, CEC, IVM, VIOS, or BCMM object connection by using the **ssh** command for the NIM client object.

When applied to SPOT resources, the **check** operation performs root synchronization for diskless and dataless clients. If required, the operation rebuilds the boot images of the SPOT network.

When applied to LPP_Source resources, the **check** operation rebuilds the contents views (**.toc**) file in the LPP_Source directory. It also determines whether all filesets are included in the resources to qualify for the LPP_Source **simages** attribute.

Using the NIM chwpar operation

Use the **chwpar** operation to change the characteristics of a WPAR. The client must have a valid mgmt_profile for the managing system, and the managing system must be running.

The command line syntax for the **chwpar** operation is as follows:

```
nim -o chwpar -a Attribute=Value ... TargetName|TargetNames
```

The target of an **chwpar** operation can be a WPAR client or group of WPAR clients.

There are no required attributes for the **chwpar** operation. The following optional attributes can be specified for the **chwpar** operation:

-a cmd_flags=Value
: Specifies flags to pass through to the operation on the managing system to activate the system. See the **chwpar** command for allowable flags.

-a group=Value
: Specifies the name of a WPAR group to use for the operation.

-a show_progress=Value
: Indicates whether status should be displayed as the operation is performed. The default value is `show_progress=yes`.

Using the NIM cust operation

Use the **cust** operation to install software filesets and updates on standalone clients and **SPOT** resources.

Note: The following operation is not allowed when resources with architectures different from the client are allocated to the client.

See "Customizing NIM clients and SPOT resources" on page 141 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 |TargetNames
```

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:

Item	Description
-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:

Item	Description
-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 while 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.
-a live_update_data=*Value*	Identifies the `live_update_data` resource that provides the AIX Live Update information for the **cust** operation.
-a live_update=*Value*	Specifies that a Live Update operation must be executed. If a `live_update_data` resource is specified, the resource is Network File System (NFS)-exported from the NIM resource server and mounted on the client. On the Network Installation Manager (NIM) client, the mounted `live_update_data` resource is copied to the `/var/adm/ras/liveupdate/lvupdate.data` file. If a `live_update_data` resource is not specified, the file that is located at `/var/adm/ras/liveupdate/lvupdate.data` on the client is used.

Using the NIM deactivate operation

Use the **deactivate** operation to stop a managed system. The client must have a valid mgmt_profile for the managing system, and the managing system must be running.

The command line syntax for the **deactivate** operation is as follows:

```
nim -o deactivate -a Attribute=Value ... TargetName|TargetNames
```

The target of a **deactivate** operation can be a WPAR client or group of WPAR clients.

There are no required attributes for the **deactivate** operation. The following optional attributes can be specified for the **deactivate** operation:

-a cmd_flags=Value
: Specifies flags to pass through to the operation on the managing system to deactivate the system. For WPAR clients, see the **stopwar** command for allowable flags.

-a group=Value
: Specifies the name of a WPAR group to use for the operation.

-a show_progress=Value
: Indicates whether status should be displayed as the operation is performed. The default value is `show_progress=yes`.

Using the NIM deallocate operation

Use the **deallocate** operation 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 |TargetNames
```

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:

Item	Description
-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.

Using the NIM define operation

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 “Setting up NIM networks” on page 185, “NIM machines” on page 108, “Using NIM resources” on page 224, and “Using NIM groups” on page 226.

Note: NIM resource class objects should not be defined with a location attribute of `/tmp` or `/tmp` subdirectories (including filesystems mounted under `/tmp`).

Using the NIM diag operation

Use the **diag** operation to prepare resources for a client to be network-booted into diagnostics mode.

Note: The following operation is not allowed when resources with architectures different from the client are allocated to the client.

The command line syntax for the **diag** operation is as follows:

```
nim -o diag -a Attribute=Value ... TargetName |TargetNames
```

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:

Item	Description
-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:

Item	Description
-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.

Using the NIM dkls_init operation

Use the **dkls_init** operation 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 |TargetNames
```

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:

Item	Description
-a paging=*Value*	Specifies the **paging** resource that contains client paging files.
-a spot=*Value*	Specifies the **SPOT** resource to be used to provide network boot support and the **/usr** file system for clients.

One of the following two attributes must be specified for the **dkls_init** operation:

Item	Description
-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** resource.
-a shared_root=*Value*	Specifies the **shared_root** resource that contains the client root (/) directories. The **shared_root** resource must have been created from the same **SPOT** resource that is specified by the **-a spot** attribute.

The following are optional attributes that can be specified for the **dkls_init** operation:

Item	Description
-a dump=*Value*	Specifies the dump resource that contains client dump files.
-a configdump=*Value*	Specifies the type of firmware-assisted dump to configure on the client. Valid values are: • selective, which avoids dumping user data. • full, which dumps all of the memory of the client partition. • none, which unconfigures the dump. The selective and full memory dumps are collected in the dump resource allocated to the client. Only POWER6 or later clients that have the appropriate firmware installed can dump to the dump resource.
-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 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.

Using the NIM dtls_init operation

Use the **dtls_init** operation 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 |TargetNames
```

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:

Item	Description
-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:

Item	Description
-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.

Using the NIM fix_query operation

Use the **fix_query** operation 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 |TargetNames
```

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:

Item	Description
-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.

Using the NIM lppchk operation

Use the **lppchk** operation 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 |TargetNames
```

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:

Item	Description
-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.

Using the NIM lppmgr operation

The NIM **lppmgr** operation helps to manage base installation images and update images in an **lpp_source**.

Although the **lppmgr** command is a separate command, it does use NIM objects as parameters. By having the **lppmgr** operation in NIM, the **lppmgr** command is called by NIM to be executed on **lpp_source** resources on other servers, and allows NIM to do sufficient checking of the **lpp_source** before and after **lppmgr** is executed. The format of the operation will be as follows:

The format of the command is as follows:

```
nim -o lppmgr -a lppmgr_flags=<flags> <lpp_source_object>
```

The NIM **lppmgr** operation is also available in SMIT using the **nim_lppmgr** fast path.

The **lppmgr** operation does not check the **lppmgr_flags** attribute for conflicts.

Note: Do not use the **-p** flag for prompting to move or remove installation images.

To list the names of duplicate filesets which should be removed with space usage information, type the following:

```
nim -o lppmgr -a lppmgr_flags="-lsb" lpp_source1
```

For more information on the **lppmgr** command, see the *Commands Reference*.

Using the NIM lswpar operation

Use the **lswpar** operation to list the characteristics of a WPAR. A WPAR client must have a valid mgmt_profile for the managing system, and the managing system must be running.

The command line syntax for the **lswpar** operation is as follows:

```
nim -o lswpar -a Attribute=Value ... TargetName|TargetNames
```

The target of a **lswpar** operation can be a WPAR client, group of WPAR clients, stand-alone client, or group of stand-alone clients. If the target is a stand-alone client or group of stand-alone clients, the operation lists information for WPARs which are on the target, regardless of whether or not those WPARs are NIM clients.

There are no required attributes for the **lswpar** operation. The following optional attributes can be specified for the **lswpar** operation:

-a cmd_flags=Value
: Specifies flags to pass through to the operation on the managing system to activate the system. See the **lswpar** command for allowable flags.

-a group=Value
: Specifies the name of a WPAR group to use for the operation.

-a show_progress=Value
: Indicates whether status should be displayed as the operation is performed. The default value is `show_progress=yes`.

Using the NIM maint operation

Use the **maint** operation to uninstall software filesets and commit and reject updates on standalone clients and **SPOT** resources.

Note: The following operation is not allowed when resources with architectures different from the client are allocated to the client.

The command line syntax for the **maint** operation is as follows:

```
nim -o maint -a Attribute=Value ... TargetName |TargetNames
```

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:

Item	Description
-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:

Item	Description
-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**.

Using the NIM maint_boot operation

Use the **maint_boot** operation to prepare resources for a client to be network-booted into maintenance mode.

Note: The following operation is not allowed when resources with architectures different from the client are allocated to the client.

The command line syntax for the **maint_boot** operation is as follows:

```
nim -o maint_boot -a Attribute=Value ... TargetName |TargetNames
```

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:

Item	Description
-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:

Item	Description
-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.

Using the NIM reboot operation

Use the **reboot** operation to reboot a NIM client machine.

The command line syntax for the **reboot** operation is as follows:

```
nim -o reboot -a Attribute=Value ... TargetName |TargetNames
```

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:

Item	Description
-a inst_warning=*Value*	Indicates whether 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.

Using the NIM remove operation

Use the **remove** operation to remove objects from the NIM environment.

The command line syntax for **remove** is as follows:

```
nim -o remove TargetName |TargetNames
```

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.

Using the NIM reset operation

Use the **reset** operation 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 |TargetNames
```

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:

Item	Description
-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. For information on deallocating resources, see "Using the NIM deallocate operation" on page 270.

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.

Using the NIM select operation

Use the **select** operation 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 |TargetNames
```

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:

Item	Description
-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.

Using the NIM showlog operation

Use the **showlog** operation 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 |TargetNames
```

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 **showlog** operation:

Item	Description
-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: **alt_disk_install** Output from the **alt_disk_install** operation **boot** Machine's boot log **bosinst** Output from the BOS installation program **devinst** Output from the installation of key system and device-driver software **liveupdate** Output from the AIX Live Update operation **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**.

Item	Description
-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.

Notes:

- The **showlog** operation has no required attributes.
- The Live Update output is located at the `/var/adm/ras/liveupdate/logs` path on the Network Installation Manager (NIM) standalone client. For additional output information during a Live Update operation, refer to the log files available in this path.

Using the NIM showres operation

Use the **showres** operation 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**, **exclude_files**, **adapter_def**, or a **live_update_data** resource.

The following are optional attributes that can be specified for the **showres** operation:

-a client=*Value*
: Specifies which client's secondary adapter configuration file is displayed from an **adapter_def** resource. This attribute is only applicable when the target of the operation is an **adapter_def** resource.

-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.

Using the NIM sync operation

The **sync** operation synchronizes the NIM database with an alternate master.

The command backs up the local NIM database, restores the database onto the alternate master, and then updates the restored database.

The command line syntax for the **sync** operation is as follows:

```
nim [-F] -o sync -a Attribute=Value ... TargetName
```

The target of a **sync** operation must be an **alternate_master**.

The following are optional attributes that can be specified for the **sync** operation:

Item	Description
-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.
-F	Specifies that NIM should **force** the operation. Use the force operation if the database on the **alternate_master** should be overwritten.

Note: There are no required attributes for the **sync** operation.

Using the NIM sync_roots operation

Use the **sync_roots** operation 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 are optional flags and attributes that can be specified for the **sync_roots** operation:

Item	Description
-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**.
-F	Specifies that NIM should **force** the operation.

A **sync_roots** operation can be performed automatically when the **check** operation is performed on a **SPOT**.

Using the NIM syncwpar operation

Use the **syncwpar** operation to synchronize the software of WPAR clients with the managing system. The client must have a valid mgmt_profile for the managing system, and the managing system must be running.

The command line syntax for the **syncwpar** operation is as follows:

```
nim -o syncwpar -a Attribute=Value ... TargetName|TargetNames
```

The target of an **syncwpar** operation can be a WPAR client, group of WPAR clients, stand-alone client, or group of stand-alone clients. If the target is a stand-alone client or group of stand-alone clients, the operation applies to the WPARs which are on the target, regardless of whether those WPARs are NIM clients.

There are no required attributes for the **syncwpar** operation. The following optional attributes can be specified for the **syncwpar** operation:

-a cmd_flags=Value
Specifies flags to pass through to the operation on the managing system to activate the system. See the **syncwpar** command for allowable flags.

-a group=Value
Specifies the name of a WPAR group to use for the operation.

-a show_progress=Value
Indicates whether status should be displayed as the operation is performed. The default value is `show_progress=yes`.

Using the NIM takeover operation

The **takeover** operation allows a machine that is configured as an **alternate_master** to take control of the NIM environment.

The alternate master attempts to become the current master of each client defined in its database by updating each client's `/etc/niminfo` file. This operation also attempts to update the database on the target **alternate_master**.

The command line syntax for the takeover operation is as follows:

```
nim [-F] -o takeover -a Attribute=Value ... TargetName
```

The target of a takeover operation must be an **alternate_master**.

The following are optional attributes that can be specified for the **takeover** operation:

Item	Description
-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.
-a show_progress *[yes | no]*	Indicates whether status should be displayed as the operation is performed. The default value is **show_progress=yes**.
-a async= *[yes | no]*	If this attribute is set to "yes", then the clients will be updated with the new master information asynchronously. The default is to run this command asynchronously.
-F	Specifies that NIM should **force** the operation. Use the force operation if the database on the **alternate_master** should be overwritten.

Note: The**takeover** operation has no required attributes.

Using the NIM unconfig operation

Use the **unconfig** operation 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 uninstalled.

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

Using the NIM update operation

The NIM **update** operation updates **lpp_source** resources by adding and removing packages.

The format of the **update** operation is as follows:

```
nim -o update -a packages=<all | list of packages with levels optional> \
   [-a gencopy_flags=<flags>] ] [-a installp_bundle=<bundle_file>] \
  [-a smit_bundle=<bundle_file>] [-a rm_images=<yes>] \
  [-a source=<dir | device | object>] [-a show_progress=<yes | no>]
  <lpp_source_object>\
```

The NIM **update** operation is also available in SMIT using the **nim_update** fast path.

The source attribute must be a directory or device that is local to the server of the target **lpp_source** resource or an existing NIM **lpp_source** resource. The default operation is to add packages to the target **lpp_source**. If the **rm_images** attribute is present, the operation will remove packages from the **lpp_source**. A user must specify either the source or **rm_images** attribute and must specify the **packages**, **installp_bundle**, or **smit_bundle** attribute, but not more than one.

Generally, the **all** keyword means to perform a multi-volume installation when the source is a CD-ROM. However, the **update** operation will only do a single volume copy (equivalent to passing the **gencopy -S** flag).

The default behavior for this command is to display output. To turn off the output, pass **show_progress=no**.

To add packages to an **lpp_source** resource, run the following:

```
# nim -o update -a packages=all -a source=/tmp/inst.images lpp_source1
```

To remove packages from an **lpp_source** resource, run the following:

```
# nim -o update -a packages="bos.games 5.1.0.25 bos.sysmgt.nim" -a rm_images=yes lpp_source2
```

Using the NIM updateios operation

The NIM **updateios** operation performs updates and customization to the Virtual I/O Server (VIOS).

The format of the **updateios** operation is as follows:

```
nim -o updateios -a Attribute=Value ... TargetName
```

The target of an **updateios** operation can be a VIOS NIM management client or an IVM NIM management client.

The following are optional attributes that can be specified for the **updateios** operation:

Attribute	Description
-a filesets=*Value*	Specifies a list of file sets to remove from the target.

Attribute	Description
-a installp_bundle=*Value*	Specifies an **installp_bundle** resource that lists file sets to remove on the target.
-a lpp_source=*Value*	Identifies the **lpp_source** resource that will provide the installation images for the **updateios** operation.
-a accept_licenses=*Value*	Specifies whether the software licenses should be automatically accepted during the installation. The default value is **accept_licenses**=no.
-a updateios_flags=*Value*	Identifies the flags that tell **updateios** what operation to perform on the VIOS. The valid values are -install, -commit, -reject, -cleanup and -remove. The default value is **updateios_flags**=-install.
-a preview=*Value*	Specifies a preview operation for the **updateios** operation. The default value is **preview**=yes.

Using EZNIM

The SMIT EZNIM feature organizes the commonly used NIM operations and simplifies frequently used advanced NIM operations.

Features of SMIT EZNIM include:

- Task-oriented menus
- Automatic resource naming that includes the level of the software used to create NIM resources.
- The user can review what steps will take place *before* executing a task, whenever possible.

Use the SMIT **eznim** fast path to open the EZNIM main menu. If the NIM environment has not been set up on your system, the EZNIM main menu displays the following options:

- Configure as a NIM Master
- Configure as a NIM client

Using EZNIM to configure a NIM master

Follow these steps to configure your current system as a NIM master.

If you select **Configure as a NIM Master**, the following options display:

```
Setup the NIM Master environment
Enable Cryptographic Authentication
Add fixes to the NIM Master environment
Add client to the NIM environment

Update clients
Backup a client
Reinstall clients
Reset clients

Show the NIM environment
Verify the NIM environment
Remove NIM environment
```

- To configure your current system as a NIM master, select **Setup the NIM Master environment**. You can select the software source to configure from, select the volume group to use for the NIM resources, and select the file system to use for the NIM resources. When the NIM master environment is configured, the basic NIM resources are created. To view the NIM resources created by EZNIM, select **Show the NIM environment**, or run the **lsnim** command on the NIM master.
- To configure your NIM master for SSL authentication, select **Enable Cryptographic Authentication**. This option allows you to install and configure the cryptographic software in the OpenSSL RPM

package. After you configure OpenSSL, NIM clients with OpenSSL installed can request cryptographic authentication during service requests from the NIM master.

- To install updates and maintenance or technology level packages to the NIM master, select **Add fixes to the NIM Master environment**. This option performs an update installation of a specified set of fixes onto the default **SPOT** resource. A second **SPOT** resource containing the newly installed fixes is created by this operation. You can optionally select to update all your NIM clients during this operation.
- To update a client using EZNIM, select **Update clients**. This option allows you to perform an **update_all** operation on a selected client (or clients) using an **lpp_source** resource.
- To back up a client using EZNIM, select **Backup a client**. This option allows you to create a system backup image of a selected client and store the backup image on the NIM master.
- To reinstall a client using EZNIM, select **Reinstall clients**. This option allows you to perform a **mksysb** restore or native, **rte** install on a selected client (or clients). You must then select a system backup image to restore or an **lpp_source** to install and decide whether to reboot and install the client now.
- To reset a NIM client to the *ready* state, select **Reset clients**. This option resets the state of a client or clients in the NIM environment. Use this option after a NIM operation has failed, and you want to return the client to the *ready* state.

Using EZNIM to configure a NIM client

Follow these steps to configure a NIM client with EZNIM.

On a client system, use the SMIT **eznim** fast path. Select **Configure as a NIM client**, and the following options display:

```
Add this system to a NIM environment
Configure Client Communication Services
Update this system
Reinstall this system
Reset this system
Show the NIM environment
```

- To define your client in the NIM environment, select **Add this system to a NIM environment**.
- To configure your NIM client for SSL authentication, select **Configure Client Communication Services**. This option allows you to install and configure the cryptographic software in the OpenSSL RPM package. After you configure OpenSSL, you can select **nimsh** as the communication protocol used by the client. Any incoming NIM master service requests are then authenticated through SSL socket connections.
- To update your client, select **Update this system**. This option allows you to perform an **update_all** operation on your client using an **lpp_source** resource.
- To reinstall your client, select **Reinstall this system**. This option allows you to perform a **mksysb** restore or native, **rte** install on a selected client (or clients). You must then select a system backup image to restore or an **lpp_source** to install and decide whether to reboot and install the client now.
- To reset your client in the NIM environment, select **Reset this system**. This option resets the state of the client in the NIM environment. Use this option after a NIM operation has failed, and you want to return the client to the *ready* state.
- To view the default resources in the EZNIM environment, select **Show the NIM environment**. The resources are defined using EZNIM Master Operations.

Example: Using EZNIM

Follow these steps to create EZNIM setup using a different volume group.

To create EZNIM setup using a different volume group, run **smitty eznim** > **Configure as a NIM Master** > **Setup the NIM Master environment**.

```
          Easy NIM - Setup the NIM Master environment

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

                                                      [Entry Fields]
  Select or specify software source                  [cd0]             +
  to initialize environment:

  Select Volume Group for resources                  [rootvg]          +

  Select Filesytem for resources                     [/export/eznim]

  Options
    CREATE system backup image?                      [yes]             +
    CREATE new Filesystem?                           [yes]             +
    DISPLAY verbose output?                          [no]              +
```

For **Select Volume Group for resources**, select your volume group. If you prefer to use a different filesystem than the default **/export/eznim**, then fill in the desired value in the **Select Filesystem for resources** field.

The **nim_master_setup** command can also be used to select a different volume group or filesystem by specifying the flag attribute (**-a file_system=<value>** or **-a volume_group=<value>**). You can enter the value for **filesystem** and **rootvg**.

To define NIM resources in using install media located in device **/dev/cd0**, and create a new filesystem named **/export/nimfs** under volume group **myvg**, type:

```
nim_master_setup -a volume_group=myvg -a file_system=/export/nimfs
```

Note: If the **/export/nimfs** filesystem does not currently exist, then is created under the **myvg** volume group. If the **myvg** volume group does not exist, it is created using the next empty physical volume (disk) since the disk attribute was not specified.

Using network installation files

The use of miscellaneous files pertinent to network installation is described.

Sample customizing script

This customizing script configures the target's TCP/IP domain name resolution and routing.

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'
```

Sample definition file for the nimdef command

This example shows 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."
```

Using the certificate viewing file

These examples are from a certificate viewing script for OpenSSL certificates.

The script is located in the **/usr/samples/nim/ssl** directory.

The script is provided for helping users view hash, issuer, subject, and other certificate information available using the **openssl** command. The script can be modified based on user need or preference.

To print out all readable values for certificate(s):

```
# certview certificate_names
```

To print out the hash value for certificate(s):

```
# certview -h certificate_names
```

To print out the issuer value for certificate(s):

```
# certview -i certificate_name
```

To print out the subject value for certificate(s):

```
# certview -s certificate_name
```

To print out the subject, issuer, and enddate values for certificate(s):

```
# certview -I certificate_name
```

Using the certificate password loading file

The following are examples from a certificate password loading file for NIM OpenSSL certificates.

The file is located in the `/usr/samples/nim/ssl` directory. The file is provided for helping users store a desired password for decrypting the NIM master's client key. The password provided must match the password used to encrypt the NIM master's client key during NIM SSL configuration.

To load the encrypted key's password in the NIM environment:

```
#  certpasswd
```

To unload the encrypted key's password from the NIM environment:

```
#  certpasswd -u
```

Only the NIM master's client key may be password encrypted. To password encrypt the NIM master's client key, complete the following steps:

1. On the NIM master, edit the `/ssl_nimsh/configs/client.cnf` config file.
2. Locate the *encrypt_key* variable and change the value to **yes**.
3. Add the *output_password* variable underneath **encrypt_key** and specify the password. If you do not specify *output_password*, you will be prompted for the password during key generation.
4. Type the following command:

   ```
   # make -f  /usr/samples/nim/ssl/SSL_Makefile.mk  client
   ```
5. On each SSL client, copy the new `server.pem` file using the **nimclient -c** command.
6. Load the password into the NIM environment using **certpasswd**.

When you use password encrypted keys, NIM commands may fail with the following error if the correct password is not loaded:

```
0042-157 nconn: unable to access the "clientkey.pem" file
```

After the password is loaded, it will be used for client key decrypting until you unload the password.

Sample KDC server definition file

Using a sample script, you can create and configure a Key Distribution Center (KDC) server on the same system as a NFS V4 server.

The script is located in the `/usr/samples/nim/krb5` directory. The script helps you create a simple KDC environment that can be modified based on your needs or preference.

Note: Before you begin, review the `config_rpcsec_server` script.

The `config_rpcsec_server` script handles the following operations:

- Creates a system user; the default is *nim*
- Creates principals for the administrator and system user
- Creates an NFS host key for the server
- Creates realm-to-domain mapping
- Creates a tar image of `krb5` files for use by KDC slim clients
- Refines the exports list
- Recycles the NFS services
- Re-exports NFS file systems and directories

Examples

To create a simple KDC environment using default values, type the following command:

```
config_rpcsec_server
```

To create a KDC environment using system user *nimadmin* as the user principal and password *l0gin1* for the **kadmin** principal, type the following command:

```
config_rpcsec_server -p l0gin1 -u nimadmin
```

Sample slim client definition file

Using a sample script, you can create and configure a NIM client as a Kerberos slim client.

The script is located in the `/usr/samples/nim/krb5` directory. The script helps you create a simple Kerberos slim client using values defined in the `config_rpcsec_server` script. The script can be modified based on user need or preference.

Note: Before you begin, review the `config_rpcsec_client` script prior to use.

The `config_rpcsec_client` script handles the following operations:

- Creates a system user; the default is *nim*
  The user must match an existing user principal on the KDC server.
- **tftp** the slim image from the NIM master
  The tar image must exist on the NIM server.
- Enables the user principal using the **kinit** command
  The password must match the user principal on the KDC server.
- Recycles the NFS services

Examples

To create a simple KDC slim client using default values, type the following command:

```
config_rpcsec_client
```

To create a simple KDC slim client using system user *nimadmin* as the user principal, type the following command:

```
config_rpcsec_client -u nimadmin
```

Troubleshooting NIM

Solutions for network boot problems and procedures for producing debug output for NIM BOS installations is described.

Refer to "NIM error and warning messages" for information about error messages.

NIM error and warning messages

Information about Network Installation Management (NIM) error and warning messages is provided, 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 nature of the problem.

All NIM error messages begin with `0042` and are followed by a three-digit error code.

Note:

1. If you require usage information for a NIM command, type the command without any parameters or with a question mark as a parameter (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. 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.
3. 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.

Information about each message listed in this chapter is organized in the following manner:

Item	Description
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

Note: 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.

Item	Description
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 Message Database located on the Information Center Web page 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.

Item	Description
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.

Item	Description
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.

Item	Description
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.

Item	Description
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.

Item	Description

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.

Item	Description

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.

Item	Description
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.

Item	Description
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 Troubleshooting in *Networks and communication management*.
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.

Item	Description
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

Item	Description
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

Item	Description
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.

Item	Description
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,**, **chrp** 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.

Item	Description
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.

Item	Description
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

Item	Description
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.
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.

Item	Description
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

Item	Description
Explanation	The group *GroupName* 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

Item	Description
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

Item	Description
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

Item	Description
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 routing`*X*`="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 routing`*X*`="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.

Item	Description
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 “NIM Commands” on page 118.
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

Item	Description
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_res** 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

Item	Description
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.

Item	Description
Message	0042-323
Explanation	To perform an operation on a NIM object, the NIM resource allocated to the object must be of the same architecture as the NIM object.
User Action	Deallocate the conflicting resource and allocate a resource with the same architecture as the object.
Message	0042-324
Explanation	Cross-platform resources and operations are not allowed on servers with an operating system level prior to AIX 5.1.
User Action	Try performing operation on a server with an operating system level of AIX 5.1 or later.
Message	0042-325
Explanation	To perform the operation, the resource and server must be of the same architecture.
User Action	Perform the operation with a server and resource of the same architecture.
Message	0042-326
Explanation	If an architecture value is specified during the creation of a cross-platform resource, then it should correctly identify the architecture of the source being used.
User Action	Give the correct architecture of the resource or do not specify an architecture when defining the resource.
Message	0042-327
Explanation	Cross-platform SPOT resources may only be created from an existing SPOT resource.
User Action	Use an existing cross-platform SPOT as the source to create the new SPOT resource.
Message	0042-330
Explanation	NIM cannot determine the architecture of the source being used for the current operation.
User Action	In an **lpp_source** resource is being created, then supply a value for the **arch** attribute.

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 network boot stages.

The network boot stages are listed in the following tasks:

Verifying network communication between the client and server:

Before initiating the network boot on the client, perform these steps to verify network communication between the client and the server.

1. 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.

   If the server can be pinged from another machine, the network adapter on the boot client may be faulty.
4. 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. For information on network-debugging procedures, refer to TCP/IP troubleshooting in the *Networks and communication management*.

Obtaining the boot image from the server:

Follow this procedure to obtain 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.

If **bootp** packets continue to be sent but not received, the boot server may not be responding to the request.

2. 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 incorrect. 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.

3. 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.

4. 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.

5. 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/inetd.conf** 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.

6. Retry the 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.

   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.

7. 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.

   After the client receives the **bootp** reply, it will **tftp** the boot image from the server.

   The number of **tftp** packets transferred to the client will be displayed at the client machine.

   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.

8. 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.

9. 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.

10. 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. Re-create 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 installation program" on page 316.

Item	Description
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.

Obtaining debug output for NIM BOS installations

Due to problems in the network or in the NIM configuration, clients may fail to boot or install properly. When this happens, it may be necessary to obtain 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 while 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 installation program. The commands and output from the BOS installation 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 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 determine 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, give special attention 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

Use these commands to create debug versions of the network boot images.

1. Use the SMIT interfaces or run the following command:

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

   where *SPOTName* is the name of your **SPOT**.
2. Obtain the address for entering the debugger by doing the following:

   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 **chrp**-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 installation program

Method A involves entering a special value at one of the installation menus and Method B uses a **bosinst_data** resource to tell the installation program to display debug output.

Both methods are described as follows:

Producing debug output without using a bosinst_data resource (Method A):

Use this procedure to produce debug output without using a bosinst_data resource.

1. To enable debugging for the BOS installation program, start by performing all the processing you would normally do to install a client.

   Because you are not using a **bosinst_data** resource, you will be prompted to supply information about the installation to the BOS installation program.
2. Select your console.
3. Select your language.
4. The **Welcome to Base Operating System Installation and Maintenance** menu is displayed. Instead of selecting one of the options, type 911 at the prompt and press Enter.
5. Continue the normal procedure for selecting options and specifying data until the installation begins. Debug output will be sent to the client's display while the installation proceeds.

Producing debug output when using a bosinst_data resource (Method B):

Use this procedure to produce debug output when using a bosinst_data resource.

1. To enable debugging for the BOS installation 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 while the installation continues.

Debugging when port number conflicts with NIM and other applications

Follow this procedure if the **nimesis** daemon will not run.

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.

Creating and installing system backups

Use the following information to create and install system backups.

Notes:

1. References to CD also apply to DVD.

2. AIX provides the **cdromd** CD and DVD automount facility, which is included in the **bos.cdmount** fileset. To determine if the cdromd daemon is enabled on your system, run the following command:

   ```
   # lssrc -s cdromd
   ```

   The **cdromd** daemon can interfere with scripts, applications, or instructions that attempt to mount the CD or DVD device without first checking to see if the device is already enabled. A resource or device busy error occurs in such a condition. Use the **cdumount** or **cdeject** command to unmount the device. Then mount the device as specified in the program or instructions. Alternatively, use the **cdcheck -m** or **mount** command to determine the current mount point of the device. For further information, see the **cdromd** command documentation in *Commands Reference, Volume 1*.

   The installation code allows for this automatic mounting. If **cdromd** is enabled and the **mkcd** command is run, the CD-R or DVD-RAM is ejected after the image is completed. If you do not want to have the media ejected, then the **cdromd** daemon should be put in the inoperative state with the following command:

   ```
   # stopsrc -s cdromd
   ```

Creating system backups

You can create and verify a bootable backup copy, or *mksysb image*, of your root volume group. You can also make separate backup copies of user volume groups.

The *root volume group* is a hard disk or group of disks that contains:

- Startup files
- Base Operating System (BOS)
- System configuration information
- Optional software products

A *user volume group*, also called the *nonroot volume group*, typically contains data files and application software.

A system backup does the following:

- Contains a working copy of your system. In the event your system data becomes corrupted, you can use this information to restore your system to working order.
- Allows you to transfer installed and configured software from one system to others. You can use the SMIT to make a backup image of the root volume group or user volume groups.

A backup transfers the following configurations from the source system to the target system:

- **rootvg** volume group information
- Paging space information
- Logical volume information
- Placement of logical partitions (if creating map files has been selected in SMIT).

  Note: The use of map files is not recommended if you plan to reinstall the backup to target systems other than the source system, or the disk configuration of the source system is to be changed before reinstalling the backup.

Using the SMIT backup menu lets you preserve configuration information, thus avoiding some of the configuring tasks normally required after restoring a system backup. A backup preserves the configuration if the following are true:

- The target system has the same hardware configuration as the source system.

  AND

- The target disk has enough space to hold the backup image.

The SMIT uses the **mksysb** command to create a backup image, stored either on CD, DVD, removable hard disk cartridge, tape, or in a file. If you choose CD, DVD, removable hard disk cartridge, or tape, the backup program by default writes a *boot image*, which makes the medium suitable for installing. For more information, see "Creating a system backup to CD-R, DVD-R, or DVD-RAM" on page 322.

If you have problems with installations from a **mksysb** image, see Troubleshooting problems with installation from mksysb backup.

Installing all device and kernel support before the backup is created

Create a system backup that contains all devices and kernel types.

All devices and kernels are installed by default when performing a base operating system installation. This allows you to create a system backup that contains all devices and kernel types. Because the system backup contains all the devices and kernel support, the system backup can be used to install another system without the need for the AIX product media. This option is available in the Install Options menu in the BOS installation menus. If you change the default (**yes**) to **no**, only the devices and kernel type for your system configuration will be installed.

This value is read from the **ALL_DEVICES_KERNELS** field in the `/var/adm/ras/bosinst.data` file on the product media that you used to boot the system.

After the system is installed, you can check if all the devices and kernel types have been installed on the system as follows:

```
# grep ALL_DEVICES_KERNELS bosinst.data
```

Output similar to the following displays:

```
ALL_DEVICES_KERNELS = yes
```

For more information about the `bosinst.data` file, refer to "Customizing your installation" on page 81.

Preparing to create system backups

Meet these prerequisites before creating system backups.

Before creating system backups, complete the following prerequisites:

- Be sure you are logged in as root user.
- Consider altering passwords and network addresses if you use a backup to make master copies of a source system. Copying passwords from the source to a target system can create security problems. Also, if network addresses are copied to a target system, duplicate addresses can disrupt network communications.
- Mount all file systems you want to back up. The **mksysb** command backs up mounted JFS (journaled file systems) and JFS2 (enhanced journaled file systems) in the **rootvg**. Refer to the **mount** command for details.

  Note: The **mksysb** command does not back up file systems mounted across an NFS network.
- Unmount any local directories that are mounted over another local directory.

  This backup procedure backs up files twice if a local directory is mounted over another local directory in the same file system. For example, if you mount **/tmp** over **/usr/tmp**, the files in the `/tmp` directory are then backed up twice. This duplication might exceed the number of files a file system can hold, which can cause a future installation of the backup image to fail.
- Use the `/etc/exclude.rootvg` file to list files you do not want backed up.
- Make at least 40 MB of free disk space available in the `/tmp` directory. The **mksysb** command requires this working space for the duration of the backup.

  Use the **df** command, which reports in units of 512-byte blocks, to determine the free space in the `/tmp` directory. Use the **chfs** command to change the size of the file system, if necessary.

For example, the following command adds 40 MB of disk space to the **/tmp** directory of a system with 4 MB partitions:

```
chfs -a size=+80000 /tmp
```

- All hardware must already be installed, including external devices, such as tape and media drives.
- The `bos.sysmgt.sysbr` fileset in the BOS System Management Tools and Applications software package must be installed. The `bos.sysmgt.sysbr` fileset is automatically installed. To determine if the `bos.sysmgt.sysbr` fileset is installed on your system, type:

  ```
  lslpp -l bos.sysmgt.sysbr
  ```

  If your system has the `bos.sysmgt.sysbr` fileset installed, continue with one of the following procedures:
  - "Creating a root volume group backup to tape, removable hard disk cartridge, or file"
  - "Creating a system backup to CD-R, DVD-R, or DVD-RAM" on page 322
  - "Backing up a user volume group" on page 326

  If the **lslpp** command does not list the `bos.sysmgt.sysbr` fileset, install it before continuing with the backup procedure. Refer to "Optional products and service updates" on page 335 for instructions, or enter the following command:

  ```
  installp -agqXd device bos.sysmgt.sysbr
  ```

  where *device* is the location of the software; for example, `/dev/cd0` for CD-ROM drive.

Creating a root volume group backup to tape, removable hard disk cartridge, or file

Follow this procedure for creating a root volume group backup to tape, removable hard disk cartridge, or file.

You can use SMIT to create a system backup to be stored to tape, removable hard disk cartridge, or in a file.

For instructions on how to back up to CD or DVD, see "Creating a system backup to CD-R, DVD-R, or DVD-RAM" on page 322.

To create a root volume group backup:

- Use the following SMIT procedure:
  1. Enter the **smit mksysb** fast path.
  2. In the Back Up the System menu, make the following selections:
     - Select which medium you want to use in the **Backup DEVICE or File** field. If you want to create a bootable backup, the medium must be tape, removable hard disk cartridge, or CD/DVD. See "Creating a system backup to CD-R, DVD-R, or DVD-RAM" on page 322 for more information. Then, select the appropriate option below:

       TAPE, removable hard disk cartridge
       Press the F4 key to list available devices and highlight the device name.

       FILE Enter a full path and file name in the entry field.
     - If you want to create map files, select **yes** in the **Create Map Files?** field.

       Map files match the physical partitions on a drive to its logical partitions. When installing from a backup image, the BOS installation program uses map files to position the logical volumes on the target drive in the same partitions they were on in the source system. If you do not create map files, the installation program relies on the logical volume manager (LVM) to determine placement for the logical volumes. For more information, see Using map files for precise allocation in *Operating system and device management*.

Note: If you plan to reinstall the backup to target systems other than the source system, or if the disk configuration of the source system might change before reinstalling the backup, do not create map files.

- To exclude certain files from the backup, select **yes** in the **Exclude Files** field, then create a `/etc/exclude.rootvg` file with an ASCII editor, and enter the file names that you do not want included in your system backup image. You can use patterns for the file names that conform to the pattern matching conventions of the **grep** command. For example, to exclude all the contents of the directory called `scratch`, put the following line in the exclude file:

  ```
  /scratch/
  ```

  For another example, exclude the contents of the directory called `/tmp` and avoid excluding any other directories that have `/tmp` in the pathname by adding the following line to the exclude file:

  ```
  ^./tmp/
  ```

  Note: All files are backed up relative to the current working directory. This directory is represented by a **.** (dot character). To exclude any file or directory for which it is important to have the search match the string at the beginning of the line, use a **^** (caret character) as the first character in the search string, followed by a **.** (dot character), and then followed by the file name or directory to be excluded.

  If the file name or directory being excluded is a substring of another file name or directory, use **^.** (caret character followed by dot character) to indicate that the search should begin at the beginning of the line and/or use **$** (dollar sign character) to indicate that the search should end at the end of the line.
- To list each file as it is backed up, select **yes** in the **List files as they are backed up?** field. Otherwise, you see a percentage-completed progress message while the backup is created.
- If you modified the `image.data` file and do not want a new one created, select **no** for **Generate new /image.data file?**. (The `image.data` file contains information about the sizes of all the file systems and logical volumes in your rootvg.)
- If you are creating a bootable backup (to tape or removable hard disk cartridge) and you want to expand the system `/tmp` file system (if required by the backup program), select **yes** for **EXPAND /tmp if needed?**.
- If the tape drive you are using provides packing (or compression), set the **Disable software packing of backup?** field to **yes**.
- If you chose tape as the backup medium, either leave the default in the **Number of BLOCKS to write in a single output** field or enter a different number.
- If you chose file as the backup medium, press Enter. If you chose tape or removable hard disk cartridge as the backup medium, insert the first blank tape or removable hard disk cartridge into the drive and press Enter.

3. The COMMAND STATUS screen displays, showing status messages while the system makes the backup image.

   If you chose tape or removable hard disk cartridge as the backup medium, the system might prompt you to insert the next tape or removable hard disk cartridge during the backup by displaying a message similar to the following:

   ```
   Mount next Volume on /dev/rmt0 and press Enter.
   ```

   If this message displays, remove the tape or removable hard disk cartridge and label it, including the BOS version number. Then insert another tape or removable hard disk cartridge and press Enter.

   When the backup process finishes, the **COMMAND:** field changes to **OK**.
4. When the backup completes, press F10 to exit SMIT.
5. If you selected tape or removable hard disk cartridge as the backup medium, remove the last tape or removable hard disk cartridge and label it. Write-protect the backup tapes or removable hard disk cartridge.

6. Record any backed-up root and user passwords. Remember that these passwords become active if you use the backup to either restore this system or install another system.

You have created the backup of your root volume group (rootvg). If you created bootable tapes or removable hard disk cartridge, you can use these tapes or removable hard disk cartridge to start your system if for some reason you cannot boot from hard disks.

Creating a system backup to CD-R, DVD-R, or DVD-RAM

Creating a backup on CD-R, DVD-R, or DVD-RAM media is similar to making a backup tape for your personal use, but with some noticeable differences.

For DVD media, the following formats for creating backups are available:

- ISO9660 CD format, which is available for DVD-R/DVD-RAM media.
- Universal Disk Format (UDF), which is available for DVD-RAM media. For information about creating a backup to DVD-RAM using UDF, see "Creating system backups using DVD-RAM media and Universal Disk Format" on page 325.

Note: For information about CD-R, DVD-R, or DVD-RAM drives and CD-R, DVD-R, or DVD-RAM creation software, refer to the following readme file:

```
/usr/lpp/bos.sysmgt/mkcd.README.txt
```

SMIT uses the **mkcd** command, which calls the **mksysb** or **savevg** command, if needed.

For system backups, the CDs or DVDs can be created as:

- Non-bootable CDs or DVDs
- Bootable CDs or DVDs

A bootable system backup contains a boot image and all the device and kernel packages necessary to install a system. A backup CD or DVD can be used to install (clone) a large number of machines, which is convenient when each machine in the system environment needs to have the same image installed.

Note: It is possible that a backup CD or DVD would not boot all machines of the same type because not every machine has the same hardware configuration. Depending on what packages were made available during the creation of the backup, the backup might not have all the necessary packages to boot an individual system. Most required packages for systems are present on the BOS AIX media.

SMIT interfaces are available for the **mkcd** command. Online help can guide you through the required steps.

Meeting hardware and software requirements for system backups:

These are the hardware and software requirements for system backup to CD-R, DVD-R, or DVD-RAM.

The **mkcd** command requires that you already have the software installed to create a CD or DVD file system in Rock Ridge format and to *burn* or write the CD or DVD. The GNU versions of the **cdrecord** and **mkisofs** commands are installed with a BOS installation. Hardware and software that have been tested with this command include the following:

Software	Hardware
GNU and Free Software Foundation, Inc. readcd command version 1.9 mkisofs command version 1.13	DVD-RAM

Preparing to run the mkcd command:

To run the **mkcd** command, you need extra working space.

A separate file system or directory is required for each of the following:

- Storing a **mksysb** or **savevg** image
- Storing the CD or DVD file system contents
- Storing the CD or DVD images before they are recorded

The **mkcd** command creates the following file systems if they are not already present or if alternative file systems or directories have not been specified:

/mkcd/mksysb_image
Space requirement depends on the size of the **mksysb** image that is to be created. The **mkcd** command attempts to calculate this space and verify that adequate space is available before starting to create the **mksysb** image.

Note: When the **mkcd** command calculates the space requirements needed for the **/mkcd/mksysb_image** directory, it also adds the space used by the excluded files (**/etc/exclude.rootvg**). It is therefore possible that the **mkcd** command might not be able to create the **/mkcd/mksysb_image** directory.

/mkcd/cd_fs
Requires 645 megabytes (up to 4.38 GB for DVD)

/mkcd/cd_images
Requires at least 645 megabytes (up to 4.38 GB for DVD) of space. If the **-R** or **-S** flags are used to specify not removing the images and there are multiple volumes required, more space must be provided.

The space used in these file systems is only temporary (unless the **-R** or **-S** flag is specified to save the images). If the **mkcd** command creates the file systems, it also removes them. Each file system or directory might require over 645 megabytes (up to 4.38 GB for DVD).

If your machine does not have sufficient space, you can use NFS to mount some space from another server system; however, the file systems must be writable. You can create a **/mkcd** file system that is very large (1.5 GB for CD or 9 GB for DVDs). The **/mkcd** file system can then be mounted onto the clients when they want to create a backup CD or DVD for their systems. When creating very large backups (larger than 2 GB) with the **mkcd** command, the file system must be large-file enabled and the **ulimit** values must be set to `unlimited`.

The **mkcd** command with the **-L** flag allows the creation of DVD-sized ISO9660 images. The **mkcd** command with the **-U** flag allows the creation of UDF DVD images. You can also use the **mkdvd** command to create DVD-sized ISO9660 images.

Creating a root volume group backup on CD or DVD with the ISO9660 format:

Follow this procedure to create a root volume group backup on CD or DVD with the ISO9660 format.

You can use SMIT to create a root volume group backup on CD or DVD with the ISO9660 format, as follows:

- To create a backup to CD, use the **smit mkcd** fast path.
- To create a backup to DVD, use the **smit mkdvd** fast path and select **ISO9660 (CD format)**.

The following procedure shows you how to use SMIT to create a system backup to CD. (The SMIT procedure for creating a system backup to an ISO9660 DVD is similar to the CD procedure.)

1. Type the **smit mkcd** fast path. The system asks whether you are using an existing **mksysb** image.
2. Type the name of the CD-R device. (This can be left blank if the **Create the CD now?** field is set to `no`.)
3. If you are creating a **mksysb** image, select **yes** or **no** for the **mksysb** creation options, **Create map files?** and **Exclude files?**. Verify the selections, or change as appropriate.

   The **mkcd** command always calls the **mksysb** command with the flags to extend **/tmp**.

   You can specify an existing `image.data` file or supply a user-defined `image.data` file. See step 16 on page 325.
4. Enter the file system in which to store the **mksysb** image. This can be a file system that you created in the **rootvg**, in another volume group, or in NFS-mounted file systems with read-write access. If this field is left blank, the **mkcd** command creates the file system, if the file system does not exist, and removes it when the command completes.
5. Enter the file systems in which to store the CD or DVD file structure and final CD or DVD images. These can be file systems you created in the rootvg, in another volume group, or in NFS-mounted file systems. If these fields are left blank, the **mkcd** command creates these file systems, and removes them when the command completes, unless you specify differently in later steps in this procedure.
6. If you did not enter any information in the file systems' fields, you can select to have the **mkcd** command either create these file systems in the **rootvg**, or in another volume group. If the default of **rootvg** is chosen and a **mksysb** image is being created, the **mkcd** command adds the file systems to the exclude file and calls the **mksysb** command with the **-e** exclude files option.
7. In the **Do you want the CD or DVD to be bootable?** field, select **yes** to have a boot image created on the CD or DVD. If you select **no**, you must boot from a product CD at the same *version.release.maintenance* level, and then select to install the system backup from the system backup CD.
8. If you change the **Remove final images after creating CD?** field to `no`, the file system for the CD images (that you specified earlier in this procedure) remains after the CD has been recorded.
9. If you change the **Create the CD now?** field to `no`, the file system for the CD images (that you specified earlier in this procedure) remains. The settings that you selected in this procedure remain valid, but the CD is not created at this time.
10. If you intend to use an Install bundle file, type the full path name to the bundle file. The **mkcd** command copies the file into the CD file system. You must have the bundle file already specified in the **BUNDLES** field, either in the `bosinst.data` file of the **mksysb** image or in a user-specified `bosinst.data` file. When this option is used to have the bundle file placed on the CD, the location in the **BUNDLES** field of the **bosinst.data** file must be as follows:

    `/../usr/sys/inst.data/user_bundles/`*`bundle_file_name`*
11. To place additional packages on the CD or DVD, enter the name of the file that contains the packages list in the **File with list of packages to copy to CD** field. The format of this file is one package name per line.

    If you are planning to install one or more bundles after the **mksysb** image is restored, follow the directions in the previous step to specify the bundle file. You can then use this option to have packages listed in the bundle available on the CD. If this option is used, you must also specify the location of installation images in the next step.
12. Enter the location of installation images that are to be copied to the CD file system (if any) in the **Location of packages to copy to CD** field. This field is required if additional packages are to be placed on the CD (see the previous step). The location can be a directory or CD device.

13. You can specify the full path name to a customization script in the **Customization script** field. If given, the **mkcd** command copies the script to the CD file system. You must have the **CUSTOMIZATION_FILE** field already set in the `bosinst.data` file in the **mksysb** image or else use a user-specified `bosinst.data` file with the **CUSTOMIZATION_FILE** field set. The **mkcd** command copies this file to the `RAM` file system. Therefore, the path in the **CUSTOMIZATION_FILE** field must be as follows:

    /../filename
14. You can use your own `bosinst.data` file, rather than the one in the **mksysb** image, by typing the full path name of your `bosinst.data` file in the **User supplied bosinst.data file** field.
15. To turn on debugging for the **mkcd** command, set **Debug output?** to `yes`. The debug output goes to the **smit.log**.
16. You can use your own `image.data` file, rather than the `image.data` file in the **mksysb** image, by typing the full path name of your `image.data` file for the **User supplied image.data file** field.

Creating system backups using DVD-RAM media and Universal Disk Format:

Universal Disk Format (UDF) allows you to manipulate files directly on the DVD-RAM media.

The system backup image is an archived file composed of many files that cannot be manipulated. However, the installation packages and any files that are not contained in the backup image, can be directly manipulated on the DVD-RAM. After the DVD is mounted the files can be changed by using an editor or new files can be copied to the DVD using the various copy and restore commands such as the **cp**, **mv**, **restore** commands.

With UDF and DVD-RAM, system space is only needed for the backup image. A high-level description of the UDF backup process is as follows:

1. Create a backup of a volume group to a file (archive) on a hard disk containing enough space to hold the backup image.
2. Populate UDF with files needed to boot and install a system.
3. Copy backup to DVD-RAM media.

The **mkcd** or the **mkdvd** command with the **-U** flag is used to create a UDF file system on the DVD-RAM.

UDF allows for the possibility of changing files directly on the DVD-RAM media, such as a `bosinst.data` file and `image.data` or `vgname.data` file. Without UDF for example, to add a user-defined `bosinst.data` file to a backup image, you must restore the backup image to a location, add the file, and then back up the files again.

Or, you had to create a supplemental diskette containing the changed `bosinst.data` file, and use the supplemental diskette in conjunction with the backup. However, some system configurations might not provide diskette drives, making this procedure more difficult.

Creating a root volume group backup on DVD-RAM with Universal Disk Format:

Use this procedure to creating a root volume group backup on DVD-RAM with Universal Disk Format (UDF).

To create a root volume group backup on DVD-RAM with UDF, do the following:

- Use SMIT to create a backup to DVD-RAM with UDF, as follows:
    1. Enter the **smit mkdvd** fast path. The system asks whether you are using an existing **mksysb** image.
    2. Select **UDF (Universal Disk Format)**.

3. Enter the name of the DVD-RAM device.
4. If you are creating a **mksysb** image, select **yes** or **no** for the mksysb creation options. The options are as follows:
   - **Create map files?**
   - **Exclude files?**

   The **mkcd** command always calls the **mksysb** command with the flags to extend **/tmp**.

   You can specify an existing `image.data` file or supply a user-defined `image.data`. See step 14.
5. Enter the file system or directory in which to store the **mksysb** image. This can be a file system you created in the **rootvg**, in another volume group, or in NFS mounted file systems with read-write access. If left blank, the **mkcd** command creates the file system and removes it when the command completes.
6. If you did not enter information in the file system field, you can select to have the **mkcd** command either create these file systems in the **rootvg**, or in another volume group. If the default of **rootvg** is chosen and a **mksysb** image is being created, the **mkcd** command adds the file systems to the exclude file and calls the **mksysb** command with the exclude files option **-e**.
7. Do you want the DVD to be bootable? If you select **no**, you must boot from a product CD at the same *version.release.maintenance* level, and then select to install the system backup from the system backup DVD.
8. If you intend to use an Install bundle file, enter the full path name to the bundle file. The **mkcd** command copies the file into the DVD file system. You must have the bundle file already specified in the **BUNDLES** field, either in the `bosinst.data` file of the **mksysb** image or in a user-specified `bosinst.data` file. When this option is used to have the bundle file placed on the DVD, the location in the `BUNDLES` field of the `bosinst.data` file must be as follows:

   `/../usr/sys/inst.data/user_bundles/`*`bundle_file_name`*
9. Additional packages can be placed on the CD by entering the name of the file that contains the packages list in the **File with list of packages to copy to DVD** field. The format of this file is one package name per line.

   If you are planning to install one or more bundles after the **mksysb** image is restored, follow the directions in the previous step to specify the bundle file. You can then use this option to have packages listed in the bundle available on the DVD. If this option is used, you must also specify the location of installation images in the next step.
10. Enter the location of installation images that are to be copied to the CD file system (if any) in the **Location of packages to copy to DVD** field. This field is required if additional packages are to be placed on the DVD (see the previous step). The location can be a directory or DVD device.
11. You can specify the full path name to a customization script in the **Customization script** field. If given, the **mkcd** command copies the script to the CD file system. You must have the **CUSTOMIZATION_FILE** field already set in the `bosinst.data` file in the **mksysb** image or use a user-specified `bosinst.data` file with the **CUSTOMIZATION_FILE** field set. The **mkcd** command copies this file to the `RAM` file system. Therefore, the path in the **CUSTOMIZATION_FILE** field must be as follows:

    `/../`*`filename`*
12. You can use your own `bosinst.data` file, rather than the one in the **mksysb** image, by entering the full path name of your `bosinst.data` file in the **User supplied bosinst.data file** field.
13. To enable debugging for the **mkcd** command, set **Debug output?** to **yes**. The debug output goes to the `smit.log`.
14. You can use your own `image.data` file, rather than the `image.data` file in the **mksysb** image, by entering the full path name of your `image.data` file for the **User supplied image.data file** field.

Backing up a user volume group

The **savevg** command provides the ability to create a user-volume group backup to a CD, DVD, removable hard disk cartridge, tape, or file.

The **savevg** command finds and backs up all files belonging to a specified volume group. The volume group must be varied-on, and the file systems must be mounted.

This user backup contains a copy of a non-rootvg volume group, and is useful for volume groups that contain user data.

The **savevg** command uses a data file created by the **mkvgdata** command. The data file created is as follows:

`/tmp/vgdata/vgname/vgname.data`

The `vgname.data` file contains information about a user volume group. The **savevg** command uses this file to create a backup image that can be used by the **restvg** command to re-create the user volume group.

The **savevg** command with the **-r** flag is used to back up only a user-volume group's logical volume structure information. The data needed to list backup properties is also backed up. The **-r** flag runs the **mkvgdata** command for the volume group specified to create a `vgname.data` file. The **-r** flag backs up only the `vgname.data` file, any map files, and the `backup.data` file. The backup image that is created is used with the **restvg -r** command option to create only the volume group, logical volumes, and file system information contained in the file, without restoring any data. For example, to back up only the *paul* user volume group's structure information to the `/vg_backup/paul_vg_data` file, type the following:

```
savevg -r -f /vg_backup/paul_vg_data paul
```

You can also use the **mkcd** command to create a user volume group backup to CD or DVD. The **mkcd** command saves one volume group at a time to a CD or DVD.

The **mkcd** command with the **-L** flag allows the creation of ISO9660 DVD sized images. The **mkcd** or the **mkdvd** command with the **-U** flag allows the creation of UDF DVD images.

If your **rootvg** image and **savevg** image are small enough to fit on one CD or DVD, you can save them both by using the **-l** (stacklist) and **-z** (customization_script) flags. The **-l** flag gives a list of images to copy to the CD or DVD. The **-z** flag lets you create a script to restore **savevg** backups. For example, if you make a copy of a non-rootvg volume group ahead of time, and then write a script that calls the **restvg** command, your non-rootvg volume group would be restored to `hdisk2` at the end of the installation of **rootvg**, as shown by the following command:

```
restvg -d /SPOT/installp/ppc/savevg_image hdisk2
```

This procedure is recommended *only* if you know you want to restore the non-rootvg volume group every time you install. Otherwise, you might just want to store it on the CD/DVD, then use **restvg** to restore it after reboot. The **restvg** command can restore from CD or DVD if the name of the image is *savevg_image*. If you save the non-rootvg backup on a CD or DVD with a different file name, you can insert that CD or DVD and use the full path to the file name as the device for the **restvg** command.

Use SMIT to back up user volume groups to CD or DVD.

Creating a user volume group backup using SMIT:

With this procedure, you can use SMIT to create a backup image of a user volume group.

1. To back up a user volume group to a tape, rdx - removable hard disk cartridge or file using SMIT, type `smit savevg` on the command line. Back up a user volume group to CD by typing `smit savevgcd` on the command line. Back up a user volume group to DVD by typing `smit savevgdvd` on the command line.
2. When the Save a Volume Group screen displays, use the steps for backing up the root volume group as a guide for backing up user volume groups. There is one exception to this procedure. If you want

to exclude files in a user volume group from the backup image, create a file named **/etc/exclude.***volume_group_name*, where *volume_group_name* is the name of the volume group you want to backup.

3. If you exclude files, edit the **/etc/exclude.***volume_group_name* file and enter the patterns of file names that you do not want included in your backup image. The patterns in this file are input to the pattern-matching conventions of the **grep** command to determine which files are excluded from the backup.

Using the user volume group backup options

After you have a system backup or a user volume group backup, you may want to verify the backup or list information about the backup image.

You can use this information for the operations you can perform on a backup image. The commands used to perform these operations are the **lsmksysb** command for system backups, and the **lssavevg** command for user volume groups. Using the **lsmksysb** command or the **lssavevg** command, you can perform the operations described in the following topics:

Previewing information about a volume group backup:

The preview option allows you to view volume group information, the date and time the backup was made, and the level of AIX.

You can use the **lsmksysb** command or the **lssavevg** command with the **-l** option to preview a backup image. For example, to preview a system backup file called `/tmp/mybackup`, type the following:

```
# lsmksysb -l -f /tmp/mybackup
```

Output similar to the following displays:

```
VOLUME GROUP:           rootvg
BACKUP DATE/TIME:       Mon Jul 29 22:03:27 CDT 2010
UNAME INFO:             AIX va08 2 5 000974AF4C00
BACKUP OSLEVEL:         7.1.0.0
none
MAINTENANCE LEVEL:      none
BACKUP SIZE (MB):       1408
SHRINK SIZE (MB):       1242

rootvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
hd5                 boot       1     1     1    closed/syncd  N/A
hd6                 paging     16    16    1    open/syncd    N/A
hd8                 jfs2log    1     1     1    open/syncd    N/A
hd4                 jfs2       1     1     1    open/syncd    /
hd2                 jfs2       21    21    1    open/syncd    /usr
hd9var              jfs2       1     1     1    open/syncd    /var
hd3                 jfs2       1     1     1    open/syncd    /tmp
hd1                 jfs2       1     1     1    open/syncd    /home
hd10opt             jfs2       1     1     1    open/syncd    /opt
fslv00              jfs2       31    31    1    open/syncd    /export/nim
fslv01              jfs2       1     1     1    open/syncd    /tftpboot
```

To preview a backup image in the SMIT, use the **lsbackupinfo** fast path.

Verifying system backup (tape only):

You can list the contents of a **mksysb** image on tape.

To list the contents of a **mksysb** image on tape, you can use SMIT (type `smit lsmksysb` on the command line). The listing verifies most of the information on the tape, but does not verify that the backup media can be booted for installations. The only way to verify that the boot image on a **mksysb** tape functions properly is by booting from the media.

Viewing the backup log for volume group and system backups:

You can view the backup log that is created each time a volume group is backed up. The log file contains information on previous volume group and system backups.

You can use the **lsmksysb** command or the **lssavevg** command with the **-B** option to view the backup log file. Type:

```
# lsmksysb -B
```

Output similar to the following displays:

```
#Device;Command;Date;Shrink Size;Full Size;Maintenance Level
/export/mksysb/generic_sysb;"mksysb -X -e /export/mksysb/generic_sysb";M
on Jul 29 22:11:17 CDT 2010;1242;1408;
/export/mksysb/generic_sysb;"mksysb -X -e /export/mksysb/generic_sys
b";Tue Jul 30 16:38:31 CDT 2010;2458;2720;
```

To view the backup log in the SMIT, select **View the Backup Log** in the System Backup Manager menu.

Viewing filesets installed in a system backup:

You can view the filesets installed in a system backup using the **lsmksysb** command with the **-L** option.

For example, to view the filesets installed in a system backup, type the following:

```
# lsmksysb -L -f generic_sysb
```

Output similar to the following displays:

```
Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
IMNSearch.bld.DBCS         2.4.0.0  COMMITTED  NetQuestion DBCS Buildtime
Modules
.
.
.
bos.terminfo.wyse.data     7.1.0.0  COMMITTED  Wyse Terminal Definitions
bos.txt.spell.data         7.1.0.0  COMMITTED  Writer's Tools Data
bos.txt.tfs.data           7.1.0.0  COMMITTED  Text Formatting Services Data
```

To view the filesets installed in a system backup in SMIT, use the **lslppbackup** fast path.

Installing system backups

You can install the Base Operating System (BOS) using a system backup image, also called an *mksysb image*.

You can use a system backup to restore a corrupted operating system. Installing a system from a backup can also reduce (or even eliminate) repetitive installation and configuration tasks. For example, you can use a backup to transfer optional software installed on the *source* system (the machine from which you created the backup copy), in addition to the basic operating system. Also, the backup image can transfer many user configuration settings to the *target* system (a different machine on which you are installing the system backup).

You can install a system from a backup image that is stored on tape, CD, or DVD, or in a file.

Note: You can boot from a DVD and use a tape for the installation. However, during a tape boot, you cannot use the CD and DVD drives to supply customized information.

The procedures to install from backup operate either in prompted or nonprompted mode, depending on conditions set in the `/bosinst.data` file and on compatibility between the backup image and the installed machine.

When installing the backup image, the system checks whether the target system has enough disk space to create all the logical volumes stored on the backup. If there is enough space, the entire backup is recovered. Otherwise, the installation halts and the system prompts you to choose additional destination hard disks.

File systems are created on the target system at the same size as they were on the source system, unless the backup image was created with **SHRINK** set to **yes** in the `image.data` file, or you selected **yes** in the BOS Install menus. An exception is the `/tmp` directory, which can be increased to allocate enough space for the **bosboot** command. If you are installing the AIX operating system from a system backup that uses the JFS file system, you cannot use a disk with 4K sector sizes.

When you install a system backup on the source machine, the installation program restores the Object Data Manager (ODM) on that machine. Restoring the ODM allows devices to keep the same number they had on the original system. For example, if you install two Ethernet cards in your source machine, first en0 in slot 3 and then en1 in slot 1, the cards are not renumbered if they are detected in reverse order when you install a system backup. When you clone a system backup, the installation program rebuilds the ODM on the target system after installing the image, so devices are renumbered. In both cases, the **rootvg** and all logical volumes have different IDs.

If you reinstall a system backup on the source machine, and the target system does not have exactly the same hardware configuration as the source system, the program might modify device attributes in the following target system files:

- All files in the `/etc/objrepos` directory beginning with "Cu"
- All files in the `/dev` directory

The settings in the bootlist are not restored. After a system backup restore, the bootlist is reset to the primary boot device.

Shared volume groups have AUTO ON set to `no`. Only user volume groups that have AUTO ON set to **yes** are imported. The reason for this is that shared volume groups might lock out other systems' access to the volume group because of installation queries on the shared volume group at the time of a mksysb restore operation.

Related concepts:

“Using a mksysb image to install the base operating system on a NIM client” on page 169
A **mksysb** installation restores BOS and additional software to a target from a **mksysb** image in the NIM environment.

“Customizing your installation” on page 81
You can customize your AIX installation. Customizing an installation requires you to edit the `bosinst.data` file and use it with your installation media.

Related information:

image.data File

Troubleshooting problems with installation from mksysb backup

Cloning a system backup

You can install a system backup on a target machine to propagate a consistent operating system, optional software, and configuration settings.

With a **mksysb** image, you can clone one system image onto multiple target systems. However, the target systems might not contain the same hardware devices or adapters, or require the same kernel as the source system. All devices and kernels are automatically installed during a BOS installation. As a result, when you create a system backup, the **mksysb** image contains all the device and kernel support. For example, you can create a system backup from *System_A* and install *System_A*'s **mksysb** image onto *System_B* without having to use product media to boot *System_B*.

If you are performing a clone installation, device information will not be restored to the target system by default. During a clone installation, the BOS installation process verifies that the **mksysb** image is from the system you are trying to install. If the target system and the **mksysb** image are different, the device information is not recovered. This behavior is determined by the **RECOVER_DEVICES** variable in the **bosinst.data** file. This variable can be set to Default, yes, or no. The following list shows the resulting behaviors for each value:

Default
No recovery of devices

yes Attempted rebuild of ODM

no No recovery of devices

Note: You can override the default value of **RECOVER_DEVICES** by selecting **yes** or **no** in the Backup Restore menu or by editing the value of the attribute in the `bosinst.data` file.

If the source system does not have the correct passwords and network information, you can make modifications on the target system now. Also, some products ship device-specific files. If your graphics adapter is different on the target system, verify that the device-specific filesets for graphics-related LPPs are installed.

Related concepts:

"Installing all device and kernel support before the backup is created" on page 319
Create a system backup that contains all devices and kernel types.

Installing a system backup on the source machine

You can use the command line to restore an operating system onto the same machine from which you created the backup.

For either interface, the following conditions must be met before beginning the procedure:

- All hardware must already be installed, including external devices, such as tape and CD/DVD-ROM drives.
- Obtain your system backup image from one of the following sources:

Item	Description
DVD	BOS DVD are created in one of the following ways: • Using the SMIT Back Up This System to CD menu. • From the command line, using the **mkcd** or **mkdvd** command.
Tape	BOS tapes, created in one of the following ways: • Using the SMIT Back Up the System to Tape/File menu. • From the command line, using the **mksysb -i** *Target* command. **Note:** If devices were removed from or replaced on the system after the backup was created, their information will be restored when you install a backup. The system shows these devices in a defined state because the ODM from the system at the time of backup is restored instead of rebuilt.
Network	The path to your backup image file. For information about installing a backup across a network, refer to Using a mksysb image to install the base operating system on a NIM client.

Note: Before you begin, select the tape or CD/DVD-ROM drive as the primary boot device. For additional information, refer to the section in your hardware documentation that discusses system management services.

Due to enhancements in the **mksysb** command, you can control how devices are recovered when you install a system backup on the source machine. This behavior is determined by the **RECOVER_DEVICES** variable in the `bosinst.data` file. This variable can be set to default, yes, or no. The following list shows the resulting behaviors for each value:

default
ODM is restored

yes ODM is restored

no No recovery of devices

Note: You can override the default value of **RECOVER_DEVICES** by selecting **yes** or **no** in the Backup Restore menu or by editing the value of the attribute in the `bosinst.data` file.

To use the command line:

1. You can use the **bootlist** command to display or change the primary boot device.
   To display the primary boot device:
   ```
   bootlist -m normal -o
   ```
   To change the primary boot device:
   ```
   bootlist -m normal rmt0
   bootlist -m normal cd0
   ```
2. Power off your machine by following these steps:
   a. Log in as the root user.
   b. Enter the following command:
      ```
      shutdown -F
      ```
   c. If your system does not automatically power off, place the power switch in the Off (`0`) position.

   Attention: Do *not* turn on the system unit until Step #install_sys_bckup_source_machine/dup00166.
3. Turn on all attached external devices. These include:
   - Terminals
   - CD or DVD drives
   - Tape drives
   - Monitors
   - External disk drives

   Turning on the external devices first is necessary so that the system unit can identify them during the startup (boot) process.
4. Insert the installation media into the tape or CD or DVD drive.
   You might find that on certain tape drive units, the tape drive door does not open while the system is turned off. If you have this problem, use the following procedure:
   a. Turn on the system unit.
   b. Insert the boot installation tape (insert Volume 1 if you received more than one volume).
   c. Turn off the system unit and wait for 30 seconds.
5. If you are not using an ASCII terminal, skip to Step 6. If you are using an ASCII terminal, use the following criteria to set the communications, keyboard, and display options.

Note: If your terminal is an IBM 3151, 3161, or 3164, press the Ctrl+Setup keys to display the Setup Menu and follow the on-screen instructions to set these options. If you are using some other ASCII terminal, refer to the appropriate documents for information about how to set these options. Some terminals have different option names and settings than those listed here.

Table 18. Communication Options

Option	Setting
Line Speed (baud rate)	9600
Word Length (bits per character)	8
Parity	no (none)
Number of Stop Bits	1
Interface	RS-232C (or RS-422A)
Line Control	IPRTS

Table 19. Keyboard and Display Options

Option	Setting
Screen	normal
Row and Column	24x80
Scroll	jump
Auto LF (line feed)	off
Line Wrap	on
Forcing Insert	line (or both)
Tab	field
Operating Mode	echo
Turnaround Character	CR
Enter	return
Return	new line
New Line	CR
Send	page
Insert Character	space

6. Turn the system unit power switch from Off (`0`) to On (`|`). The system begins booting from the backup media. If your system is booting from tape, it is normal for the tape to move back and forth. If your system has an LED display, the three-digit LED should display `c31`.

   Note: You can boot from production media (tape or CD) if your backup media fails to boot. The initial Welcome screen includes an option to enter a maintenance mode in which you can continue the installation from your backup media. Refer to Troubleshooting an installation from a system backup for more information.

   If you have more than one console, each terminal and directly attached display device (or console) might display a screen that directs you to press a key to identify your system console. A different key is specified for each terminal displaying this screen. If this screen is displayed, then press the specified key *only* on the device to be used as the system console. (The system console is the keyboard and display device used for installation and system administration.) Press a key on only one console.

   Note: If the **bosinst.data** file lists a valid display device for the **CONSOLE** variable, you do not manually choose a system console. Read Customizing your installation for more information about the **bosinst.data** file.
7. The type of installation that begins is determined by the settings of the **PROMPT** field in the control_flow stanza of the **bosinst.data** file. Use the following criteria to determine the type of

installation you will be using:

Item	Description
PROMPT = no	Non-prompted Installation. This installation method is used if the backup image is configured to install automatically, without having to respond to the installation program. Go to step 8.
PROMPT = yes	Prompted Installation. This installation method is used if you need to use menu prompts to install the backup image. Also, use this installation method if a non-prompted installation halts and the Welcome to Base Operating System Installation and Maintenance screen displays. Go to step 9.

8. A successful non-prompted installation requires no further instructions because the installation is automatic.

   Note: If the backup image holds source system-configuration information that is incompatible with the target system, the non-prompted installation stops and a prompted installation begins.

   The Installing Base Operating System screen displays before the installation starts. The non-prompted installation pauses for approximately five seconds before beginning. After this time, the non-prompted installation continues to completion.

   However, if you decide to interrupt the automatic installation and start a prompted session, type 000 (three zeros) at the terminal and follow the remaining steps in this procedure.
9. The Welcome to the Base Operating System Installation and Maintenance screen displays.

   Note: You can view Help information at each screen of this installation process by typing 88.

   Choose the **Change/Show Installation Settings and Install** option.
10. The System Backup Installation and Settings displays. This screen shows current settings for the system. An ellipsis follows the disk listed in the first line if there is more than one disk selected.
11. Either accept the settings or change them. For more information on using map files, see Creating system backups.

    To accept the settings and begin the installation, skip to step 16.

    To change the settings, continue with step 12.
12. Type 1 in the System Backup Installation and Settings screen to specify disks where you want to install the backup image. The Change Disk(s) Where You Want to Install screen displays. This screen lists all available disks on which you can install the system backup image. Three greater-than signs (>>>) mark each selected disk.

    Type the number and press Enter for each disk you choose. Type the number of a selected disk to deselect it. You can select more than one disk.

    Note: You can also specify a supplemental disk by typing 66 and pressing the Enter key for the **Disks not known to Base Operating System Installation** option. This option opens a new menu that prompts for a device support media for the supplemental disk. BOS installation configures the system for the disk and then returns to the Change Disk(s) Where You Want to Install screen.
13. After you have finished selecting disks, press the Enter key.

    The screen that displays after you press the Enter key is dependent on the availability of map files for *all* of the selected disks. The criteria for this is as follows:

    - If one or more selected disks have no maps, BOS installation returns directly to the System Backup Installation and Settings screen. Skip to step 15.
    - If all selected disks have maps, the Change Use Maps Status screen displays, where you choose whether to use maps for installation. Continue with step 14.

    To preserve the placement of logical volumes during a future restoration of the backup, you can create map files before backing up a system. Map files, stored in the **/tmp/vgdata/rootvg** directory, match the physical partitions on a drive to its logical partitions. Create map files either with the SMIT Backup the System menu, using the **-m** option when you run the **mksysb** command.

    For more information about map files, see Using Map Files for Precise Allocation in *Operating system and device management*.

14. Type either 1 or 2 in the Change Use Maps Status screen to specify whether the installation program is to use maps.

    When you complete this choice, BOS installation returns to the System Backup Installation and Settings screen.

15. Decide whether BOS installation is to shrink file systems on the disks where you install the system. When you choose this option, the logical volumes and file systems within a volume group are re-created to the minimum size required to contain the data. This reduces wasted free space in a file system.

    File systems on your backup image might be larger than required for the installed files. Press the 2 key to toggle the **Shrink File Systems** option between **Yes** and **No** in the System Backup Installation and Settings screen. The default setting is **No**.

    Note: Shrinking the file system disables the use of maps.

16. Type 0 to accept the settings in the System Backup Installation and Settings screen.

    The Installing Base Operating System screen displays the rate of completion and duration.

    If you specified a supplemental disk in step 12, an untitled screen temporarily replaces the Installing Base Operating System screen. When this screen displays, it prompts you to place the device-support media in the drive and press the Enter key. BOS installation re-configures the supplemental disk, then returns to the Installing Base Operating System screen.

    The system reboots automatically when the installation completes.

Optional products and service updates

After the base operating system (BOS) is installed, you might want to install optional software or service updates.

Note: The **cdromd** CD and DVD automount facility, which is included in the **bos.cdmount** fileset, is provided in AIX. To determine if the **cdromd** daemon is enabled on your system, run the following command:

```
# lssrc -s cdromd
```

The **cdromd** daemon can interfere with scripts, applications, or instructions that attempt to mount the CD or DVD device without first checking to see if the device is already enabled. A resource or device busy error occurs in such a condition. Use the **cdumount** or **cdeject** command to unmount the device. Then mount the device as specified in the program or instructions. Alternatively, use the **cdcheck -m** or **mount** command to determine the current mount point of the device. For further information, see the **cdromd** command documentation in the *Commands Reference*.

The installation code allows for this automatic mounting. If **cdromd** is enabled and the **mkcd** command is run, the CD-R or DVD-RAM is ejected after the image is completed. If you do not want to have the media ejected, then the **cdromd** daemon must be put in the inoperative state with the following command:

```
# stopsrc -s cdromd
```

Optionally installed software

What constitutes optionally installed software is described.

Optionally installable software includes:

- **Optional Software Products**: Software that is not automatically installed on your system when you install the BOS. Software products include those shipped with the operating system and those purchased separately. The BOS is divided into subsystems that can be individually updated, such as **bos.rte.install**. Any update that begins with **bos.rte** updates a BOS subsystem.

- **Service Updates**: Software that corrects a defect in the BOS or in an optional software product. Service updates are organized by filesets. This type of update always changes part of a fileset.

Software products can be divided into the following categories:

Licensed Program
A licensed program (LP) is also known as a *licensed program product* (LPP) or a *product*. An LP is a complete software product including all packages associated with that licensed program. For example, **bos** (the base operating system) is a licensed program.

Package
A group of separately installable units that provide a set of related functions. For example, **bos.net** is a package.

Fileset An individually installable option. Filesets provide a specific function. An example of a fileset is **bos.net.nfs.client 7.1**. For more information on fileset packaging, see "Fileset installation packages" on page 401.

Fileset Update
An individually installable update. Fileset updates either enhance or correct a defect in a previously installed fileset.

Bundle
A collection of packages, products, or individual filesets that suit a specific purpose, such as providing personal productivity software or software for a client machine in a network environment. A set of bundles is provided with BOS that contain a specific set of optional software. For more information on bundle packaging, see "Packaging software bundles" on page 403.

A product can be composed of several packages, which in turn can be composed of different filesets. A product might be installed in its entirety, or only certain packages or filesets for the product might be installed. Software products are subdivided in this way, because many software products are large and have many pieces that can be used independently. Dividing a product into separately installable filesets allows you to install only those filesets you need.

You can install all the filesets included in a package or the entire product, or you can install only *selected* filesets, especially if you have limited hard disk space on your system.

Identifying software products

The product name, level number, and product identification fields are described.

The product name and level number identify a software product. The format for a software product level in AIX is as follows:

versionnumber.releasenumber.modificationlevel.fixlevel

Each field in the software product identification is defined as follows:

- The *versionnumber* field consists of 1 to 2 digits that identify the version number.
- The *releasenumber* field consists of 1 to 2 digits that identify the release number.
- The *modificationlevel* field consists of 1 to 4 digits that identify the modification level.
- The *fixlevel* field consists of 1 to 4 digits that identify the fix level.

For example, `07.01.0000.0000` is a software product level number, and `07.01.0000.0032` is a software product update level. It is not necessary to include the leading zeroes in the version, release, modification level, and fix level fields of the level. Level `07.01.0000.0000` can also be written as `7.1.0.0`.

Software licensing

The types of software licensing that can be implemented in the software purchase are run-time licensing and acceptance of software license agreements.

Normally, software requiring run-time licenses is only selected for installation when you have a license to use that software. Although the System Management Interface Tool (SMIT) allow you to install licensed software even if you do not own a license, you might be prevented from using the newly installed software until you have obtained the appropriate license.

Accepting software license agreements requires that the license agreement be accepted as part of the installation process. If software installed as part of your BOS installation requires accepting a software license agreement, you cannot exit the Configuration Assistant (or the Installation Assistant for non-graphics consoles) until the license agreement has been accepted. You can view as well as accept the license agreement. The BOS installation can be customized to automatically accept software licenses. For more information, refer to "Customizing your installation" on page 81.

For optional software installation, you can preview the license agreements on the installation media using the **smit license_on_media** fast path or the **installp -El** command. During the installation process, you can use the menu item to accept the software license, or you can use the **installp** command with the **-Y** flag. To view accepted license agreements on a system, you can use the SMIT **smit installed_license** fast path or the **lslpp -E** command. When a product is uninstalled, the license agreement acceptance is changed to the inactive state. If the product is reinstalled, you will not be prompted to reaccept the license agreement.

After completing the prerequisites in the next section, your next step is deciding whether to install software with SMIT. Descriptions of both applications are included in this chapter.

Before you install optional software and service updates, refer to the specific instructions that accompany your installation media. If you ever need to reinstall your system, refer to the installation media instructions.

> **Note:** For information about developing software products that are installed using the **installp** command, refer to Packaging Software for Installation in *General Programming Concepts: Writing and Debugging Programs*.

Managing AIX editions

A unique IBM Tivoli License Manager (ITLM) signature file exists for each supported edition (standard, or enterprise). Signature files are included in the **bos.rte** subsystem and are shipped to the `/usr/lpp/bos/editions` directory. Changing the edition modifies the signature file located in the `/usr/lib/bos/swidtag` directory. Depending on the level of the AIX operating system installed, previous locations were `/usr/lpp/bos/iso-swid`, `/usr/lpp/bos/properties/version`, and `/usr/lpp/bos`.

The **chedition** command provides both a command line and SMIT interface (use fastpath **smitty editions**) to change the ITLM signature file on the system or list the current edition being run on the system. To change the edition of the system, the **chedition** command can be run with a **-s** (standard) , or **-e** (enterprise) option, and an optional **-d** (device) flag which allows for the installation of an edition bundle file. A new ITLM signature file will be copied from the `/usr/lpp/bos/editions` directory to the `/usr/lib/bos/swidtag` directory. Depending on the level of the AIX operating system installed, previous locations were `/usr/lpp/bos/iso-swid`, `/usr/lpp/bos/properties/version`, and `/usr/lpp/bos`.

If the optional **-d** flag is used, then the **chedition** command will call **geninstall** to install the content of any edition bundle files that exist, utilizing the default installp flags of **acNgX**.

If other flags are desired, the Install Software Bundle SMIT menus (fastpath **smitty install_bundle**) should be used. The **chedition** command also supports a **–p** (preview) flag. The optional **–d** flag allows an edition bundle to be installed at a later time, as an edition change is not required to install an edition bundle.

After successfully completing an edition change, if a previous edition of a bundle file exists, then an informational message appears reminding the user to remove any software that was specific to the previous edition.

If the **geninstall** command returns a non-zero value while attempting to install an edition bundle file, an error message appears and the system edition will not be updated. By default, the system edition is set to **standard**.

Preparing to install optional software products and service updates

The prerequisites for installing optional software or service updates are described.

If either of the following conditions applies to you, go to the referenced section. Otherwise, continue with the procedures in this chapter.

- If you need to commit updates or remove previously installed software, go to “Maintaining optional software products and service updates” on page 343.
- If you are using a network installation server, refer to “Network Installation Management” on page 107.

Complete the prerequisites

Before installing optional software or service updates, complete the following prerequisites:

- You must be logged in to the system as the root user.
- AIX BOS must be installed on your system. If the BOS is not yet installed on your system, go to “Installing the Base Operating System” on page 39, or if you are installing over a network, refer to Installing with Network Installation Management.
- Either insert the media that contains the optional software or service updates into the appropriate drive or know the local or routed path to the software.
- If you are installing service updates and do not have a current backup of your system, use the procedures in “Creating system backups” on page 318. To create a system backup, you must have the backup fileset (**bos.sysmgt.sysbr**) installed on your system.
- If system files have been modified, back them up separately before updates are applied, because the update process might replace configuration files.
- If you are installing from CD or DVD and have a mounted documentation disk in the same media drive that you want to install from, run the following commands in the sequence shown:

  ```
  # unlinkbasecd
  # umount /infocd
  ```

- To eject the documentation disk, press the eject button on the media drive for at least two seconds.

Checking fileset build dates

The **installp** command has been enhanced to check the "build date" of filesets being installed to ensure that an older fileset is not installed on top of a new fileset.

For example, using the sample build dates shown below (0723 represents the 23rd week of the year 2007), a Technology Level 7 fileset at level 5.3.7.0 is prevented from installing on top of Technology Level 6 fileset at level 5.3.0.80 even though 5.3.7.0 has a higher VRMF (Version, Release, Modification, Fix). Previously, only a VRMF comparison needed to be run to determine installation eligibility. Now the "build date" of the installed fileset is checked to verify that the fileset to be installed is not older.

```
YYWW    0723            0746            0816
-----------------------------------------------
TL7                     5.3.7.0         5.3.7.10
TL6     5.3.0.60        5.3.0.70        5.3.0.80
```

The following is an example of an error message from the **installp** output:

```
+-----------------------------------------------------------------------------+
                    BUILDDATE Verification...
+-----------------------------------------------------------------------------+
Verifying build dates...
0503-465 installp: The build date requisite check failed for fileset bos.rte.install.
Installed fileset build date of 0816 is more recent than the selected fileset build date of 0746.
installp: Installation failed due to BUILDDATE requisite failure.
```

Related information:

Service and support best practices

Installing optional software products or service updates

Optional software products and service updates can be installed using system management tools provided with the operating system.

After the service updates are applied and committed by the method of your choice, if a system restart is required, you can perform the AIX Live Update operation to eliminate the reboot requirement.

To view the files as they are being installed, do the following:

- In SMIT, you can set the **DETAILED Output** field to `yes` to list the files being restored during an installation.
- You can also use the **installp** command with the verbose option **(-V2)** to show which files have been updated.

Related concepts:

"Live Update" on page 376
Starting with AIX Version 7.2, the AIX operating system provides the AIX Live Update function that eliminates the workload downtime that is associated with AIX system restart that is required by previous AIX releases when fixes to the AIX kernel are deployed. The workloads on the system are not stopped in a Live Update operation, yet the workloads can use the interim fixes after the Live Update operation.

Installing optional software and service updates using SMIT

Use SMIT to install optional software and service updates.

The following installation paths are available in SMIT:

Install Software
: Install or update software from the latest levels of software available on the media. To shorten the list of software displayed, message and locale software are omitted from the list. To use this option, type `smit install_latest` on the command line.

Update Installed Software to Latest Level
: Update all currently installed software to the latest level available on the installation media. To use this option, type `smit update_all` on the command line.

Update Software to the Latest Level (Live Update)
: Beginning with AIX 7.2 Technology Level 1, you can perform the same operation as the `smitty update_all` command, except that a Live Update operation is performed, and this operation does not require a system reboot. To use this option, enter `smit lu_update_all` from the command line.

  All updates to the system must be committed before you perform the Live Update operation. All updates that are applied during the operation are committed, the file systems are expanded if

necessary, and additional requisites are installed. This operation requires a completed /var/adm/ras/liveupdate/lvupdate.data file, except when you use the **Preview** option. The output from the installation part of the operation is available in the /var/adm/ras/install_all_updates.log file.

Install Software Bundle
Install complete bundles of software simply by specifying the input device and which bundle you are installing. You can also preview a bundle installation to see what software will be installed and how much space is required in the file system to install the bundle. To use this option, type smit install_bundle on the command line.

Update Software by Fix
Install a specific fix for a problem. This menu allows you to list all service fixes on the media and select a fix to install. You can also preview the installation to see what software will be updated and how much space is required in the file system to apply the fix. To use this option, type smit update_by_fix on the command line.

Install and Update from ALL Available Software
Install or update software from all software available on the media. To use this option, type smit install_all on the command line.

The following option is available in the Install Software, Install Software Bundle, and Install and Update from ALL Available Software SMIT menus:

```
INVOKE live update?                         no
  Requires /var/adm/ras/liveupdate/lvupdate.data.
```

If you change this value to yes, the SMIT fast path runs the **geninstall** command with the **-k** flag to start the Live Update operation. If interim fixes are being installed, it must be marked as LU CAPABLE. You can use the **Preview** option to determine whether an interim fix is marked as LU CAPABLE. In AIX® 7.2 Technology Level 1, or later, you can select updates for installation when you perform a Live Update operation. If you are installing updates, you are responsible for having a viable backup of the system. Before the Live Update operation starts, you must commit all existing updates on the system. Any new updates that are installed during the Live Update operation will be committed.

To use the Live Update operation, the bos.liveupdate.rte fileset must be installed, and the /var/adm/ras/liveupdate/lvupdate.data file must be available. For more information about the lvupdate.data file, see the /var/adm/ras/liveupdate/lvupdate.template file.

Note: If a problem occurs during the installation of optional software that causes the installation process to halt abnormally, you might have to complete a *cleanup* procedure to remove the partially installed software from the system before attempting to reinstall it. If the system instructs you to do a cleanup, go to "Cleaning up optional software products and service updates" on page 345

Some installed software must ship new installation images instead of service updates in new technology levels or service packs of the AIX operating system. For instance, a new installation image is required if the requisites of the installation image changes. When filesets are updated by using the **smitty update_all** or **install_all_updates** command, the most current version of the fileset is installed irrespective of whether filesets are updated by using the installation image or service update in the software source.

When a new installation image is installed, the history of the fileset in the system, which is the output of the **lslpp -ah <fileset>** command, is reset. The output of the **lslpp -ah <fileset>** command lists the new level of the fileset instead of original installation that was installed and all changes after that installation. . The following examples show the history of the **bos.ecc_client.rte** file before and after the installation image is installed.

- Before a new installation image for the **bos.ecc_client.rte** file is shipped, the following output is displayed:

```
# lslpp -ah bos.ecc_client.rte
       Fileset         Level       Action       Status       Date       Time
       ----------------------------------------------------------------------------
       Path: /usr/lib/objrepos  bos.ecc_client.rte
                       6.1.9.0     COMMIT       COMPLETE     04/26/17     16:49:31
                       6.1.9.0     APPLY        COMPLETE     04/26/17     16:49:31
                       6.1.9.15    APPLY        COMPLETE     04/26/17     21:02:55
                       6.1.9.45    APPLY        COMPLETE     04/27/17     08:11:05
       Path: /etc/objrepos  bos.ecc_client.rte
                       6.1.9.0     COMMIT       COMPLETE     04/26/17     16:49:42
                       6.1.9.0     APPLY        COMPLETE     04/26/17     16:49:42
                       6.1.9.15    APPLY        COMPLETE     04/26/17     21:03:07
                       6.1.9.45    APPLY        COMPLETE     04/27/17     08:11:19
```

- After a new installation image for the **bos.ecc_client.rte** file is shipped and installed on the system, the following output is displayed:

```
# lslpp -ah bos.ecc_client.rte
       Fileset         Level        Action       Status       Date         Time
       ----------------------------------------------------------------------------
Path: /usr/lib/objreposbos.ecc_client.rte
                       6.1.9.100    COMMIT       COMPLETE     04/27/17     09:19:12
                       6.1.9.100    APPLY        COMPLETE     04/27/17     09:19:12

Path: /etc/objreposbos.ecc_client.rte
                       6.1.9.100    COMMIT       COMPLETE     04/27/17     09:19:22
                       6.1.9.100    APPLY        COMPLETE     04/27/17     09:19:22
```

Related concepts:

"Live Update" on page 376
Starting with AIX Version 7.2, the AIX operating system provides the AIX Live Update function that eliminates the workload downtime that is associated with AIX system restart that is required by previous AIX releases when fixes to the AIX kernel are deployed. The workloads on the system are not stopped in a Live Update operation, yet the workloads can use the interim fixes after the Live Update operation.

Completing the SMIT installation and reading the status messages

The system activity and actions that you must take after the installation process has begun is described.

Perform the following steps:

1. When you press Enter to start the installation, the COMMAND STATUS screen displays. As the installation proceeds, a series of messages display. The amount of time that the installation takes varies depending on your system and the software you are installing and updating.

   Note: The system might prompt you to insert the volume of the installation media, with a message similar to the following:

   ```
   Mount volume 2 on /dev/cd0.
   Press the Enter key to continue.
   ```

   When this message displays, insert the specified media and press Enter.

   When the installation finishes, the **Command: status** field on the COMMAND STATUS screen changes to **OK** or **failed**. **OK** indicates that the installation ran to completion, although some filesets may not have installed successfully. The **failed** status means that there was a problem with the installation. Although a preview installation always finishes with an **OK** status, always check the summaries.

   For information about error messages, refer to "Interpreting installation-related system and error messages" on page 95.
2. When the installation halts or finishes, the screen returns to the top of the list of messages that display during installation. You can review the message list as described in the next step, or you can exit SMIT and review the **smit.log** file (**/smit.log** or **/home/***user_id***/smit.log**).
3. Review the message list for error messages on software products or service updates that may not have been successfully installed. Use the following procedure to correct any errors in the installation:

a. Look at the pre- and post-installation summaries at the end of the message list to see whether any installation failure occurred.
b. Use the message list to determine problems and which software products or service updates were involved. For example, space limits might have been exceeded or the requisites might not have been met for some software. The system lists how much extra space is needed and which requisite software products or service updates to install.
c. Any product that is marked as *FAILED*, *BROKEN*, or *CANCELLED* can be reinstalled after the condition that caused the failure has been corrected. You do not need to reinstall any service update or software product that was marked as *SUCCESS* in the Installp Summary report. If you need to perform the installation again, change installation settings as appropriate. For example, if requisites were missing, set **AUTOMATICALLY install requisite software?** to **yes**. If there was not enough space to complete the installation, set **EXTEND file systems if space needed?** to **yes**.

   If you need to install again and you have AIX BOS multivolume media, insert volume 1 of the AIX product DVDs. Press F3 to return to the previous screen, then restart the installation. See "Interpreting installation-related system and error messages" on page 95 for information about **bosboot** command errors that may occur while the installation program is running, and about recovery procedures for these errors.

   Attention: If the system log files show the following message, indicating that a reboot is required, perform the reboot as indicated in Step 6:

   ```
   * * *  A T T E N T I O N  * * *
   System boot image has been updated. You should reboot the
   system as soon as possible to properly integrate the changes
   and to avoid disruption of current functionality.
   ```
d. If the installation was interrupted (for example, a power failure), you might need to use the cleanup procedure before continuing. Press F10 (or Esc+0) to exit SMIT, and refer to "Cleaning up optional software products and service updates" on page 345.
e. If the software has been installed successfully, and you have no other software to install, go to Step 4.

   If you have additional software to install from a different installation media, remove the media that is in that drive and insert the new media.

   Press F3 (or Esc+3) to return to the previous screen and continue installing the software product or service update.

4. Press F10 (or Esc+0) to exit SMIT.
5. Remove all installation media from the drives.
6. When you are directed, reboot your system by typing: `# shutdown -Fr`

Updating installed software from the command line

The **install_all_updates** command updates installed system software to the latest level that is on the media, and verifies the current recommended technology level.

Beginning in AIX 5L Version 5.2 with the 5200-01 Recommended Maintenance package, if you select the option to install all devices and kernels during a BOS installation, then during subsequent **update_all** processing, any new `devices.*` filesets are installed from the installation media. This option can be turned off by setting the **ALL_DEVICES_KERNELS** variable in the `/var/adm/ras/bosinst.data` file to `no`.

If the **ALL_DEVICES_KERNELS** variable is set to `no`, the **install_all_updates** command does *not* install any filesets that are present on the installation media but not installed on the system, unless these filesets are installed as requisites of other selected filesets.

For **installp** images, all **installp** requisites are enforced.

The following example shows how to install all **installp** updates on the **/dev/cd0** device and to verify the current recommended technology level:

```
# install_all_updates -d /dev/cd0
```

For more information about the **install_all_updates** command, refer to the *Commands Reference*.

Checking modifications to configuration files

The **geninstall** command provides an easy way to see what modifications have been made to the configuration files listed in `/etc/check_config.files`.

When these files have been changed during a **geninstall** installation or update operation, the differences between the old and new files is recorded in the `/var/adm/ras/config.diff` file. If `/etc/check_config.files` requests that the old file be saved, the old file can be found in the `/var/adm/config` directory. The `/etc/check_config.files` file can be edited and used to specify whether old configuration files that have been changed should be saved (indicated by `s`) or deleted (indicated by `d`), and has the following format:

```
d /etc/inittab
```

Maintaining optional software products and service updates

During and after installation, the following major maintenance actions can be taken with optional software products and service updates.

Whether a particular action can be taken depends on whether the action is being applied to the entire software product, or only to a service update that has had a previous action taken on it.

You can perform these actions using System Management Interface Tool (SMIT) or by using commands directly from the command line. The following sections briefly describe how to do each action using SMIT, or a command. SMIT provides an online help to guide you through each process.

Note: Any library or executable program updated by an interim fix or service update which is in use by an active process will not be reflected in that process unless it is restarted. For example, an update that changes the ksh will not have the changes reflected in any ksh processes that are already running. Likewise, an update to the `libc.a` library will not be reflected in any process that is already running. In addition, any process that is using a library and does a **dlopen** operation of the same library after the library has been updated could experience inconsistencies if it is not restarted.

Applying a service update

When installing a service update, it can be left in the *applied* state.

In this state, the former version of that software product is saved in the **/usr/lpp/***PackageName* directory. Service Updates in the applied state allow you to restore the former version of the software without having to reinstall it.

Only service updates can be placed in the applied state. In contrast, after you install an entire software product, the product is left in the *committed* state. Software products in the committed state do not save the previous version of the software, because two versions of the same software product cannot be installed at the same time.

Applying a service update using SMIT:
Type `smit update_by_fix` on the command line.

Applying a service update from the command line:
Use the **installp -a** command to only apply the update.

Rejecting a service update

When you reject an applied service update, the update files are removed from the system and the previous version of the software is restored.

Only service updates in the applied state can be rejected. You can use SMIT to reject applied service updates.

Rejecting a service update using SMIT:
Type `smit reject` on the command line.

Rejecting a service update from the command line:
Use the `installp -r` command to reject an applied update.

Attention: After the reject completes, if the system log files show the following message, indicating that a reboot is required, perform the reboot as soon as possible:

```
* * *  A T T E N T I O N  * * *
System boot image has been updated. You should reboot the
system as soon as possible to properly integrate the changes
and to avoid disruption of current functionality.
```

Removing a software product

When you remove a software product, that product's files are removed from the system, and the Software Vital Product Data information is changed to indicate that the product is removed.

The remove process also attempts to restore the system's configuration to its previous state, although this is dependent on the product and might not always be complete. After a product is removed, no version of that product remains running on the system.

Use SMIT to remove software products. If you set the **Remove dependent software?** field to **yes**, any requisite software (software that is dependent on the product you are removing) is also removed, unless it is required by other software on your system.

Removing a software product using SMIT:
Type `smit remove` on the command line.

Removing a software product from the command line:
Use the `geninstall -u` command to remove the product.

Copying a software bundle to the hard disk for future installation

The Copy Software Bundle to Hard Disk for Future Installation option allows you to copy a software bundle from a specified source to a location on your local system.

Installation software bundles include the following:

- Alt_Disk_Install
- App-Dev
- CC_EVAL.Graphics
- CDE
- GNOME
- Graphics
- KDE
- Kerberos_5
- openssh_client
- openssh_server
- PerfTools
- SbD.Graphics
- Server
- SystemMgmtClient
- Trusted_AIX

- Trusted_AIX_SYSMGT

Cleaning up optional software products and service updates

The cleanup procedure attempts to delete items that were partially installed or left in an incomplete state. For example, after an update is interrupted, the **lslpp -l** command might report the update status as **APPLYING** rather than **APPLIED**.

Note: This procedure applies only to the update or installation of optional software products. If your AIX BOS installation was unsuccessful, see "Troubleshooting a system that does not boot from the hard disk" on page 91 for more information.

The cleanup procedure attempts to revert the update to its previous state. For example, when cleaning up an update that was interrupted in the **COMMITTING** state, the cleanup procedure attempts to return the update to its **APPLIED** state.

If the interruption occurs during the initial state of an installation, then the cleanup procedure attempts to delete the installation entirely and restore the previous version of the product (if there is one). When the previous version is restored, it becomes the active version. When the previous version cannot be restored, the software is listed by the **lslpp -l** command as **BROKEN**.

When the product is deleted or **BROKEN**, you can attempt to reinstall the software. Any product in the **BROKEN** state cannot be cleaned up; it can only be reinstalled or removed.

The system automatically initiates a cleanup when an installation fails or is interrupted. Normally, you must initiate a cleanup procedure if the system shuts down or loses power during an installation or if the installation process terminates abnormally. Occasionally, you are prompted to reboot (restart) the system after running the cleanup procedure.

If you get a message indicating that no products were found that could be cleaned up, you may have run the cleanup procedure when it was not needed. Try your installation again.

If you get a message indicating that you need to clean up a failed installation, contact your point of sale for assistance.

Initiating a cleanup procedure using SMIT

Follow these steps for initiating a cleanup procedure using SMIT.

1. Type `smit maintain_software` on the command line.
2. Select **Clean Up After Failed or Interrupted Installation**.

Initiating a cleanup procedure from the command line

Perform this step to initiate a cleanup procedure from the command line.

Type `installp -C` on the command line.

Managing existing installp image source

The **lppmgr** command is used to manage an existing **installp** image source.

The **lppmgr** command performs the following functions on an existing **installp** image source (also known as an **lpp_source** resource in the NIM environment):

- Remove duplicate updates (**-u** flag).
- Remove duplicate base levels (**-b** flag).
- Eliminate update images that are the same level as base images of the same fileset. Such update images can create conflicts that lead to installation failure (**-u** flag).
- Remove message and locale filesets other than the language you specify (**-k** flag).

- Remove superseded filesets (**-x** flag).
- Remove non-system images from a NIM **lpp_source** resource (**-X** flag).

By default, **lppmgr** lists all images filtered by the preceding routines. The **-r** flag can be used to remove the filtered images and the **-m** flag can be used to move the images to another location.

The **lppmgr** command does not replace the **bffcreate** command, perform installations, or work with installed filesets. Before using the **-X** flag, it is recommended that you have a good understanding of NIM, system images (known as SIMAGES in NIM), and the workings of a NIM **lpp_source** resource.

To list all duplicate and conflicting updates in the **/myimages** image source directory, type:

```
# lppmgr -d /myimages -u
```

To remove all duplicate and conflicting updates in the **/myimages** image source directory, type:

```
# lppmgr -d /myimages -u -r
```

For more information about the **lppmgr** command, refer to the *Commands Reference*.

Using the Software Service Management menu (including SUMA)

The Software Service Management menu allows access to Service Update Management Assistant (SUMA) functions, which significantly simplify the system update process by allowing policy-based automatic downloads of technology level updates from the Web.

The Software Service Management menu allows generation of reports to manage filesets installed on a system, filesets contained in a repository, and filesets available from the IBM System p support website. It also provides a way for you to clean up and rename software images in a repository.

You can perform these actions using either the SMIT **service_software** fast path or by using commands directly from the command line. The following sections briefly describe how to do each action using SMIT or a command.

Using the Service Update Management Assistant (SUMA)

The Service Update Management Assistant (SUMA) helps move system administrators away from the task of manually retrieving maintenance updates from the Web.

SUMA offers flexible options that let you set up an automated interface to download fixes from a fix distribution website to your systems. Because SUMA can be configured to periodically check the availability of specific new fixes and entire maintenance levels, the time spent on such system administration tasks is cut significantly.

SUMA can be accessed through the **suma** command or through the SMIT **suma** fast path.

Using the SUMA command line interface:

The **suma** command can be used to perform these operations on a SUMA task or policy.

- Create
- Edit
- List
- Schedule
- Unschedule
- Delete

An *RqType* parameter specifies the type of download that is being requested (such as Technology Level (TL), Service Pack(SP), Maintenance Level (ML), or Latest). A policy can be set up to retrieve the following types of fixes:

PTF Specifies a request to download a PTF. An example is U813941. Only certain PTFs may be downloaded as an individual fileset. For example, PTFs containing the **bos.rte.install**, **bos.alt_disk_install.rte**, or PTFs that come out in between Service Packs. Otherwise, the TL or SP must be downloaded.

ML Specifies a request to download a specific maintenance level (such as **5300-11**).

TL Specifies a request to download a specific technology level (such as **6100-03**).

SP Specifies a request to download a specific service pack (such as **6100-02-04**).

Latest Specifies a request to download the latest fixes. This *RqType* value returns the latest service pack of the TL specified in FilterML.

Several flag options can be used with the **suma** command to further specify your request. With these command options, you can perform the list, edit, create, schedule, unschedule, and delete operations on different tasks or policies.

For example, to create and schedule (**-s**) a task that downloads the latest fixes on the 15th day of every month at 2:30 a.m. (using cron format), and add a policy label through the *DisplayName* field (useful when listing policies through SMIT), type:

```
suma -s "30 2 15 * *" -a RqType=Latest \
-a DisplayName="Latest fixes - 15th Monthly"
```

The preceding example uses task defaults, which can be displayed by the **suma -D** command.

Type the following command to create and schedule a task that downloads the entire 6100-03 Technology Level into the **/lppsrc/6103** directory on Monday at 11:00 p.m., runs an **lppmgr** clean operation after the download to remove any superseded updates, duplicates base levels, and conflicting updates:

```
suma -s "0 23 * * 1" -a Action=Clean -a RqType=ML -a RqName=6100-03 \
-a DLTarget=/lppsrc/6103 -a FilterSysFile=/dev/null
```

Note: Prior to running a task that specifies `Action=Clean`, you can run **suma -c** to verify the SUMA global configuration settings that will be used when running **lppmgr**. Setting **REMOVE_SUPERSEDE**, **REMOVE_DUP_BASE_LEVELS**, and **REMOVE_CONFLICTING_UPDATES** to yes will result in the intended action of the preceding example.

For a more complete listing of examples that detail the functionality of the **suma** command, refer to the **suma** command.

Using the Comparison Reports menu

The Comparison Reports menu allows you to generate several comparison reports to verify that the filesets for a particular fix or preventive maintenance package are installed by comparing filesets installed on a system to another source. This source could be a fix repository, such as an **lpp_source** or a directory of fixes, or a downloaded list from the IBM System p support Web site.

If you want to verify that your **lpp_source** is up to date, you can also compare a fix repository to a downloaded list.

You can perform these actions in the SMIT **compare_report** fast path or using the **compare_report** command.

Using the Compare Installed Software to Fix Repository menu:

The Compare Installed Software to Fix Repository menu allows you to compare the filesets installed on a system to a fix repository.

The following report lists are generated:
- Filesets on the system that are back-level (**lowerlevel.rpt**)
- Filesets on the system that are at a later level (**higherlevel.rpt**)
- Filesets in the fix repository that are not installed on the system (**notinstalled.rpt**)
- Filesets installed on the system that are not in the fix repository (**no_update_found.rpt**)

The Compare Installed Software to Fix Repository option is available using the SMIT **instofix_compare** fast path or the **compare_report** command with the following options:

```
compare_report -s -i FixDir {[-l] [-h] [-m] [-n]} [-t ReportDir -Z | -v]

compare_report -b BaseList -i FixDir {[-l] [-h] [-m] [-n]} [-t ReportDir] -Z | -v]
```

When using the **-l** (lower) or **-h** (higher) flags, the compare report only shows that interim fixes are installed. The higher or lower concept is not currently available.

Compare Installed Software to List of Available Updates menu:

The Compare Installed Software to List of Available Updates menu allows you to compare the filesets installed on a system to a downloaded list of available updates from the IBM System p service Web site.

The following report lists are generated:
- Filesets on the system that are back-level from the latest (**lowerthanlatest1.rpt**)
- Filesets on the system that are at a later level from the latest maintenance and technology levels (**higherthanmaint.rpt**)
- Filesets on the system that are back-level from the latest maintenance and technology levels (**lowerthanmaint.rpt**)

The Compare Installed Software to List of Available Updates option is available using the SMIT **instolist_compare** fast path or the **compare_report** command with the following options:

```
compare_report -s -r ServiceReport {[-l] [-h]} [-t ReportDir -Z | -v]

compare_report -b BaseList -r ServiceReport {[-l] [-h]} [-t ReportDir] -Z | -v]
```

When using the **-l** (lower) or **-h** (higher) flags, the compare report only shows that interim fixes are installed. The higher or lower concept is not currently available.

Compare Fix Repository to List of Available Updates menu:

The Compare Fix Repository to List of Available Updates menu allows you to compare the filesets in a fix repository, such as a fix directory or **lpp_source**, to a downloaded list of available updates from the IBM System p service Web site.

The report list that is generated contains information on filesets in the fix directory that are back-level from latest (**lowerthanlatest2.rpt**).

The Compare Fix Repository to List of Available Updates option is available using the SMIT **fixtolist_compare** fast path or the **compare_report** command with the following options:

```
compare_report -i FixDir -r ServiceReport [ -t ReportDir -Z | -v ]
```

Compare a list of installed software on a base system to another system:

The compare a list of installed software on a base system to another system option allows you to compare the filesets installed on a system to another system.

The **lslpp -Lc** output from one system is saved to a file and compared with the **lslpp -Lc** output from another system. The following report lists are generated:

- A list of base system installed software that is at a lower level (**baselower.rpt**)
- Filesets not installed on the base system, but installed on the other system (**otheronly.rpt**)
- A list of base system installed software that is at a higher level (**basehigher.rpt**)
- Filesets installed on the base system that are not installed on the other system (**baseonly.rpt**)

To compare a list of installed software on a base system to another system use the **compare_report** command with the following options:

```
compare_report -b BaseList -o OtherList {[-l] [-h] [-m] [-n]} [-t ReportDir -Z | -v]
```

Using the Rename Software Images in Repository option

The Rename Software Images in Repository option allows you to rename updates that have FIX ID numbers for names, to more meaningful fileset names like those generated when updates are copied to hard disk for future installation. This action renames all filesets in the indicated directory with the same format.

This option is available using the SMIT **rename_software** fast path.

You can also use the **bffcreate** command to rename software images in a directory. To rename software images in a directory using the **bffcreate** command, use the **-c** flag and the **-d** flag for the directory containing the filesets. For example, to rename filesets in the `/usr/sys/inst.images` directory, type:

```
# /usr/sbin/bffcreate -cd /usr/sys/inst.images
```

You can also create a log file containing a mapping between the old names and new names, using the **-s** *logfile* option, as shown in the following example:

```
# /usr/sbin/bffcreate -cd /usr/sys/inst.images -s /usr/sys/inst.images/names.log
```

This example creates a `/usr/sys/inst.images/names.log` file that contains content formatted as follows:

```
old_fileset_name:new_fileset_name
```

This option is also available in SMIT Rename Software Images in Repository menu as the **LOG software name changes (location of log file)** option.

Using the Clean Up Software Images in Repository option

The Clean Up Software Images in Repository option allows you to remove unneeded or duplicate software images from a local software-image repository.

You can remove duplicate software, superseded updates, and language software:

- The Remove Duplicate software option allows you to remove duplicate base and update images from the specified directory.
- The Remove Superseded updates option allows you to remove superseded filesets from the specified directory. This action applies only to update images.
- The Remove Language software option allows you to remove language and locale filesets that are not needed on your system. This option removes all language and locale filesets from the specified directory, except the language specified in the PRESERVE language field. By default, the value of the **LANG** environment variable for the system is used to determine the language to preserve.

- The Save Removed files option allows you to save all removed files to the location specified in the **DIRECTORY for storing saved files** field. Select true in this field if you want to move the images to another location instead of removing them from the hard disk drive.

This option is available using the SMIT **cleanup_software** fast path.

Using InstallShield MultiPlatform

Some products that are distributed for installation on AIX are packaged and installed with InstallShield MultiPlatform (ISMP).

Unlike **installp** or RPM Package Manager (RPM) installations which only provide non-prompted or silent installations of a product, ISMP-packaged products provide both interactive and silent interfaces for installing and un-installing a product.

Similar to products packaged and installed with **installp** and RPM, ISMP-packaged products can be installed using the AIX system management tools, including SMIT. These tools use the **geninstall** command to install or uninstall products that are packaged and installed with **installp**, RPM, or ISMP. As expected, the **geninstall** command can be used directly to install, list, or uninstall ISMP-packaged products.

For instructions for installing or un-installing a specific product packaged and installed with ISMP, consult the product's documentation.

Installing products with InstallShield MultiPlatform

You install an InstallShield MultiPlatform product using SMIT, the **geninstall** command, or the files provided by the product.

- Use the SMIT **install_software** fast path to install ISMP products without knowledge of the exact location of the product installation files. For information on installing optional software using SMIT, see “Preparing to install optional software products and service updates” on page 338. Use the F4 key on the **SOFTWARE to install** field to select the product you want to install. ISMP products are displayed in the list similar to **installp** packages or RPM packages. Select the ISMP products, and press Enter to begin the installation.

  By default, ISMP product installations launched through SMIT is *silent* or *nonprompted* installations. To perform an interactive installation, use the **geninstall** command, or the instructions provided with the product documentation.

  Although SMIT has a preview option, this option is not available for ISMP installations. If you select the preview option, a message instructs you to launch an interactive installation using the command line, which allows you to view the preinstallation summary panel before completing the product installation.
- Use the **geninstall** command to install an ISMP-packaged product. To perform an interactive installation, specify the device or directory containing the product installation files with the **-d** flag and specify the product name. The product name is the same as the subdirectory name containing the product installation files. For example, if we have a product called `MyProduct`, and the product installation files are in the `/usr/sys/inst.images/ismp/ppc/MyProduct/` directory, use the following command for an interactive installation:

  ```
  /usr/sbin/geninstall -d /usr/sys/inst.images J:MyProduct
  ```

  Use the **J:** prefix to inform the **geninstall** command that the product is an ISMP package. The **geninstall** command recognizes the `ismp/ppc` subdirectory, just as it recognizes **RPMS/ppc** for RPM packages and **installp/ppc** for **installp** packages, so it is only necessary to pass the `/usr/sys/inst.images` base directory. You can also use the directory that contains the installation files. In this example, specify the directory as follows:

  ```
  /usr/sbin/geninstall -d /usr/sys/inst.images/ismp/ppc/MyProduct J:MyProduct
  ```

  If you want to launch a *silent* or *nonprompted* installation with **geninstall**, include the **-Z** flag:

  ```
  /usr/sbin/geninstall -d /usr/sys/inst.images -Z J:MyProduct
  ```

For more information about silent installations, see "Performing a silent installation using response files."

- You can use the installation files provided by the product developer to install an ISMP-packaged product. The product developer might provide a script or executable that can be used to launch an ISMP-packaged product installation. For more information, refer to the documentation provided with the product.

Uninstalling a InstallShield MultiPlatform product

You uninstall an ISMP product using SMIT, the **geninstall** command, or the files provided by the product's developer.

- You can use the SMIT **remove** fast path to uninstall an ISMP-packaged product. If you use the F4 key to list the installed software for the **SOFTWARE to remove** field, the ISMP-packaged product is displayed in the list. You can also type the name of the product in the field.

  By default, uninstallation processes performed in SMIT are *silent* or *nonprompted*. To perform an interactive uninstallation, use the **geninstall** command, or the instructions provided with the product documentation.

  In SMIT, the preview option is not available for the ISMP product uninstallation procedure. If you attempt to preview the uninstallation, a message instructs you to launch an interactive uninstallation using the command line. This allows you to view the pre-unstallation summary panel before completing the product uninstallation.
- You can use the Software Application to uninstall ISMP-packaged products.

  Note: If you select the preview option, but proceed through the entire uninstallation wizard, the product is un-installed. Most ISMP products include a pre-uninstallation summary panel that provides preview information about the uninstallation. If you do not want to proceed with the installation after viewing this information, press the **CANCEL** button to exit the wizard.
- You can use the **geninstall** command to perform an uninstallation for an ISMP-packaged product. To perform the uninstallation interactively, specify the **-u** flag for uninstallation, and the product name. For example, to uninstall the *MyProduct* product, type the following:

  ```
  /usr/sbin/geninstall -u J:MyProduct
  ```

  To speed processing, use the **J:** prefix to inform the **geninstall** command that you are un-installing an ISMP-packaged product.

  To perform a *silent* or *nonprompted* uninstallation with the **geninstall** command, use the **-Z** flag, as follows:

  ```
  /usr/sbin/geninstall -Zu J:MyProduct
  ```
- You can use installation files provided by the product developer to uninstall an ISMP-packaged product. The product developer might provide instructions for performing an ISMP-packaged product uninstallation. For more information, see the documentation provided with the ISMP product.

Performing a silent installation using response files

You can perform silent installations for ISMP-packaged products using response files.

A response file contains predetermined responses for an installation. By default, the **geninstall** command searches on the product media in the ISMP-product subdirectory for response files for each ISMP product. For example, the `MyProduct` ISMP product subdirectory is similar to the following:

/basedir/ismp/ppc/MyProduct/

The **geninstall** command searches in the ISMP-product subdirectory for each ISMP product specified in the install list or bundle for a ***MyProduct*.response** file. If a ***MyProduct*.response** file does not exist or is not found, **geninstall** proceeds with whatever defaults are configured in the installer.

The **-t** *ResponseFileLocation* option allows you to specify an alternate location for response files or response file templates. The *ResponseFileLocation* can either be a file or directory name. If the *ResponseFileLocation* is a directory, it must already exist. If the *ResponseFileLocation* is not an existing directory, it is assumed that a file name is specified.

To use response files with ISMP products, the following methods are available:

- Create a response file template. To create an ISMP response file template in the default location, use the **geninstall** command with the **-T** flag. The **-T** flag creates an ISMP response file template in the default location, which is the directory containing the product installation files. The resulting template can be used to create a response file for future installations of the same product with the desired options. Creation of the response file template does not result in installation of the ISMP product.

  To create an ISMP response file template for the `MyProduct` ISMP product using the product installation files in the `/usr/sys/inst.images/ismp/ppc/MyProduct/` default directory, do the following:

  ```
  /usr/sbin/geninstall -d /usr/sys/inst.images -T J:MyProduct
  ```

  The `MyProduct.template` response file template that is generated is similar to the following:

  ```
  ##################################################################
  #
  # InstallShield Options File Template
  #
  # Wizard name: Setup
  # Wizard source: setup.jar
  # Created on: Tue Jun 25 10:59:55 CDT 2004
  # Created by: InstallShield Options File Generator
  #
  # This file can be used to create an options file (i.e., response file) for the
  # wizard "Setup". Options files are used with "-options" on the command line to
  # modify wizard settings.
  #
  # The settings that can be specified for the wizard are listed below. To use
  # this template, follow these steps:
  #
  #    1. Enable a setting below by removing leading '###' characters from the
  #    line (search for '###' to find settings you can change).
  #
  #    2. Specify a value for a setting by replacing the characters <value>.
  #    Read each settings documentation for information on how to specify its
  #    value.
  #
  #    3. Save the changes to the file.
  #
  #    4. To use the options file with the wizard, specify -options <filename>
  #    as a command line argument to the wizard, where <filename>  is the name
  #    of this options file.
  #
  ##################################################################
  ##################################################################
  #
  # My Product Install Location
  #
  # The install location of the product. Specify a valid directory into which the
  # product is  installed. If the directory contains spaces, enclose it in
  # double-quotes. For example, to install the product to C:\Program Files\My
  # Product, use
  #
  #    -P installLocation="C:\Program Files\My Product"
  #
  ### -P installLocation=<value>
  ```

  Although the preceding is a simple example, products often have many user-configurable options that might be set in the response file. Each of these options is presented in the template with an explanation of the expected value for that option.

- Create a response file recording. To create a response file recording, use the **geninstall** command with the **-E** flag. The **-E** flag creates an ISMP response file recording in the default location, which is the directory containing the product installation files. This option requires running the ISMP installation interactively and completely. Creation of the response file recording will also result in installation of the ISMP product.

  To record the `MyProduct.response` response file with the `MyProduct` ISMP product and the product installation files in the `/usr/sys/inst.images/ismp/ppc/MyProduct/` default directory, do the following:

  ```
  /usr/sbin/geninstall -d /usr/sys/inst.images -E J:MyProduct
  ```

  This starts the interactive installation wizard. It is necessary to run the wizard to completion to successfully create the response file recording. When completed, a message similar to the following displays:

  ```
  Options record mode enabled - run the wizard to completion to create the options file response.file
  ```

  The resulting file `MyProduct.response` response file is similar to the following:

  ```
  ############################################################
  #
  # InstallShield Options File
  #
  # Wizard name: Setup
  # Wizard source: setup.jar
  # Created on: Tue Jun 25 11:05:34 CDT 2002
  # Created by: InstallShield Options File Generator
  #
  # This file contains values that were specified during a recent execution of
  # Setup. It can be used to configure Setup with the options specified below when
  # the wizard is run with the "-options" command line option. Read each setting's
  # documentation for information on how to change its value.
  #
  # A common use of an options file is to run the wizard in silent mode. This lets
  # the options file author specify wizard settings without having to run the
  # wizard in graphical or console mode. To use this options file for silent mode
  # execution, use the following command line arguments when running the wizard:
  #
  #    -options "record.txt" -silent
  #
  ##############################################################################
  ##############################################################################
  #
  # My Product Install Location
  #
  # The install location of the product. Specify a valid directory into which the
  # product is  installed. If the directory contains spaces, enclose it in
  # double-quotes. For example, to install the product to C:\Program Files\My
  # Product, use
  #
  #    -P installLocation="C:\Program Files\My Product"
  #
  -P installLocation="/opt/MyProduct"
  ```

  The **-P installLocation** value has been completed according to the response given while running the wizard. In the preceding example, the `/opt/MyProduct` directory was specified as the installation location in the wizard. The response file generated by this action can be used directly to launch a silent installation with the chosen installation location.

- Use a response file for a silent installation. You can use a response file generated by the two methods mentioned previously or one provided with the product to perform a silent installation with the desired options.

  To use a response file for a silent installation with the **geninstall** command, the `MyProduct` product, and the installation files and response file in the `/usr/sys/inst.images/ismp/ppc/MyProduct/` default directory, do the following:

  ```
  /usr/sbin/geninstall -Zd /usr/sys/inst.images J:MyProduct
  ```

To use a response file for a silent installation with the **geninstall** command, `MyProduct` product, installation files in `/usr/sys/inst.images/ismp/ppc/MyProduct/` directory, and the `/tmp/MyProduct/MyProduct.response` response file, do the following:

```
/usr/sbin/geninstall -Zd /usr/sys/inst.images  \
      -t /tmp/MyProduct/MyProduct.response J:MyProduct
```

Using response files with NIM

If you are using NIM to install an ISMP-packaged product on one or more NIM clients, you can create and use a separate response file for each client.

Separate response files are useful when properties of the installation operation must be configured differently for each client. In order to install multiple clients, you must name each response file *CLIENT_NAME***.response**. These response files must be located in the default location (the same location as the product installer files).

For example, to install the **MyProduct** ISMP-packaged product located in an **lpp_source** resource in the `/export/lpp_source/lpp_source1/ismp/ppc/MyProduct` directory on the **CLIENT1** and **CLIENT2** clients, do the following:

1. Create a **CLIENT1.response** and **CLIENT2.response** response file.
2. Place the response files in the `/export/lpp_source/lpp_source1/ismp/ppc/MyProduct` directory.
3. Create the correct responses for each client in the corresponding response file.
4. When you run the NIM **cust** operation to install the **MyProduct** ISMP-packaged product on **CLIENT1** and **CLIENT2**, the response files are used automatically and appropriately for each client.

If you want to use the same response file for all clients, name the response file *PRODUCT_NAME***.response** and place in the same default location as the ISMP-packaged product (the product location in the **lpp_source** resource). For example, create a response file called **MyProduct.response** in the `/export/lpp_source/lpp_source1/ismp/ppc/MyProduct/` directory. If there are no client response files when you perform the NIM **cust** operation, the `MyProduct.response` file is used automatically.

Interim fix management solution

You can use the interim fix management solution to track and manage interim fix packages on a system.

An interim fix package might be an interim fix, debug code, or test code that contains commands, library archive files, or scripts that run when the interim fix package is installed.

The interim fix management solution consists of: the interim fix packager (**epkg**) command and the interim fix manager (**emgr**) command.

The **epkg** command creates interim fix packages that can be installed by the **emgr** command. The **emgr** command installs, removes, lists, and verifies system interim fixes.

Note: When the term *package* is used, **installp**'s reference is the term *fileset*.

Installing and managing interim fix packages

You can install and manage packages created with the **epkg** command.

The **epkg** command installs and manages packages created with the **epkg** command, and maintains a database with interim fix information on the system. The **emgr** command performs the following operations:

The interim fix package display:

The levels of information on the Interim fix package display are described.

The **emgr** command's **-d** flag displays the contents and topology of the efix package. The **-d** flag works with the **-v** (verbosity) flag. The default verbosity level is 1, but you can set the level to 1, 2, or 3. The syntax for interim fix package display is as follows:

```
emgr -d -e interim fixPackage | -f ListFile [-w Directory] [-v{1|2|3}]
```

For example, to get a level 1 verbosity output on the interim fix package **test.102403.epkg.Z**, type the following command.

```
# emgr -d test.102403.epkg.Z
```

The verbosity levels include the following information:

LEVEL 1

Lists one interim fix per line with the following information:

- Label for the interim fix package
- Interim fix files contained in the package
- Target location for each interim fix file

LEVEL 2

Lists the following information:

- All LEVEL 1 information
- Abstract
- Reboot requirement (`yes` or `no`)
- Prerequisite files needed
- Pre-install script
- Post-install script
- Pre-remove script
- Post-remove script
- File type for each interim fix file

LEVEL 3

Lists the following information:

- All LEVEL 2 information
- Packaging date for each interim fix file
- Virtually unique ID (VUID) for each package
- File size for each interim fix file
- Checksum for each interim fix file
- Package for each interim fix file
- Description of each interim fix file
- Contents of installation scripts and control files, if they are readable text
- Reboot scenario for each interim fix file
- Prerequisites of interim fix file on other interim fix files
- Packages that will be locked when the interim fix is installed
- Interim fixes that will be superseded when the interim fix is installed

Note: Displaying is not supported in the original release of interim fix management. You should update to the latest level of interim fix management to enable this feature. To update interim fix management, update **bos.rte.install** to the latest level.

Installing interim fix packages:

The **emgr** command installs interim fix packages that are created with the **epkg** command.

The following example shows the syntax for installing an interim fix package:

```
emgr -e interim fixPackage | -f ListFile [-w Directory] [-b] [-k] [-p] \
[-I] [-q] [-m] [-o] [-X] [-a path]
```

The interim fix package installation operation consists of the following phases:

The installation preview phase:

These steps occur during the installation preview phase.

1. The interim fix manager initializes all commands and libraries, and extracts the interim fix metadata from the interim fix package.
2. The interim fix attributes and description are listed.
3. The **emgr** command performs a lock-checking procedure by checking the installed interim fix data if the target files that are being delivered by this interim fix package already have existing interim fixes installed. If one or more interim fix files that are delivered by this interim fix package are locked, the **emgr** command does not allow the installation or installation preview to proceed.
4. The **emgr** command performs **installp** package prerequisite verification. If the user supplied an **installp** prerequisite file, the **emgr** command checks the prerequisites at this stage. If one or more of the prerequisites is not met, the **emgr** command does not allow the installation or installation preview operation to proceed.
5. The **emgr** command performs interim fix to interim fix prerequisite verification. The **emgr** command checks the following:
   - All interim fix prerequisites for the interim fix package being checked are installed.
   - All interim fix xrequisites for the interim fix package being checked are not installed.
   - There are no previously installed interim fixes that list the interim fix being checked as an xrequisite.

   Note: You can use the **epkg** command to specify the interim fix to interim fix prerequisites for an package. For more information on checking prerequisites, see "Interim fix user-specified package components" on page 369.
6. The **emgr** command checks for space requirements by checking whether the target file systems contain adequate space for the installation of the interim fix package. This includes space for unpacking the interim fix files, creating database entries, saving replaced files, installing interim fix files, creating interim fix mounts when using the **-m** flag, archiving library members, and other miscellaneous tasks. The **emgr** command also adds a small buffer to the various space calculations to account for file metadata and other factors.

   If the user specifies the auto-expand flag using the **-X** flag, then the **emgr** command attempts to expand the file system to the required size. If space requirements cannot be met, the **emgr** command stops the installation. If the user specifies a preview installation using the **-p** flag, then the **emgr** command only reports the space statistics without attempting expansion

   If the user specifies a preview installation using the **-p** flag, the **emgr** command does not perform the interim fix installation phase. Instead the **emgr** command skips to the summary and cleanup phase of installation.

The installation phase:

During the installation phase for installing interim fix packages, these steps occur.

1. During the interim fix installation setup step, the entire interim fix package is unpacked and the installation tools are initialized.

2. The **emgr** command checks whether the interim fix package supersedes any other currently installed interim fix files. If any of the installed interim fix files should be superseded, the **emgr** command removes them.

   Note: You can use the **epkg** command to specify that an interim fix package be superseded when you install another interim fix package. For more information on superseding, see "Interim fix user-specified package components" on page 369.
3. If a **pre_install** script is specified, it is run. If the **pre_install** script returns a failure, the **emgr** command halts the installation. If the **pre_install** script succeeds, the **emgr** command proceeds with the installation and sets the interim fix state to `INSTALLING`.
4. Any files that are replaced by interim fix files in the interim fix package are saved to a secured directory. If the interim fix package does not deliver any files, this step is skipped.

   From this point forward, any major failure causes the **emgr** command to run a failure-cleanup procedure, which attempts to clean up the failed installation. If this process fails, the interim fix is placed into the `BROKEN` state.
5. All interim fix files are installed to their target locations. If the installation is a mount installation operation using the **-m** flag, then the **emgr** command creates a unique mount file within the parent directory of the target file. The target file is then over-mounted by the interim fix mount point. For more information about the mount installation operation, see "Performing an interim fix mount installation operation" on page 362.
6. Package locking occurs. The interim fix package locks are processed. If the installer for which the interim fix package is created supports interim fix package locking, the **emgr** command locks the package associated with the interim fix files installed in step 4. For example, the **installp** command supports interim fix locking, so an interim fix created for an **installp** package will support interim fix package locking.

   Note: In addition to implicit locking, you can use **epkg** to specify that a certain package be explicitly locked when you install another package. For more information on locking, see "Interim fix user-specified package components" on page 369.
7. If a **post_install** script is specified, it is run. If the **post_install** script returns a failure, the **emgr** command halts the installation.
8. Reboot processing occurs. If the interim fix package specifies that a reboot operation is required, the **emgr** command issues a message to the user and makes any necessary changes to the boot image. The **emgr** command does *not* reboot the system automatically.
9. At this point, all installation steps have succeeded and the **emgr** command changes the interim fix state to `STABLE` for a standard installation operation, or `MOUNTED` for a mount installation operation.

The summary and cleanup phase:

These steps occur during the summary and cleanup phase.

1. The **emgr** command displays a summary of all operations and results. If more than one interim fix package was specified with an input file using the **-f** flag, the **emgr** command provides a report for each interim fix package.
2. The **emgr** command cleans up any temporary directories and files. It also unloads any memory modules that have been loaded into memory.

Removing an interim fix package:

The interim fix removal operation removes an installed interim fix.

You can specify an individual interim fix by using one of the interim fix identification methods or specifying several individual interim fixes by using a list file. For more information about the interim fix identification methods, see "Referencing interim fixes" on page 363.

The syntax for removing an installed interim fix is as follows:

```
emgr -r -L Label | -n interim fixNumber | -u VUID | -f ListFile [-w Directory] \
[-a path] [-b] [-k] [-p] [-I] [-q] [-X]
```

The removal preview phase:

These steps occur during the removal preview phase.

1. The interim fix manager initializes all commands and libraries, and loads interim fix metadata from the interim fix database.
2. The interim fix attributes and descriptions are listed.
3. Space requirements are checked. The **emgr** command checks whether the target file systems contains adequate space to restore the saved files. This includes space-changing database entries, restoring saved files, archiving library members, and other miscellaneous tasks. The **emgr** command also adds a small buffer to the various space calculations to account for file metadata and other factors.

   If the user specifies to auto-expand the file system using the **-X** flag, the **emgr** command attempts to expand the file system to the required size. If space requirements cannot be met, the **emgr** command halts the remove operation. If the user specifies a preview installation operation using the **-p** flag, then the **emgr** command only reports the space statistics without attempting to expand the file system.

   If the user specifies a preview installation using the **-p** flag, the **emgr** command does not perform the interim fix removal and skips to the summary and cleanup phase.

The removal phase:

These steps occur in the removal phase.

Note: Any failure in the removal phase causes the interim fix state to change to `BROKEN`.

1. The **emgr** command initializes all remove utilities and changes the interim fix state to `REMOVING`.
2. Package unlocking occurs. All packages that are locked by the interim fix file being removed are unlocked. Because it is possible that a single package may be locked by multiple interim fixes, the **emgr** command only unlocks a package if this interim fix file is the last (or the only) interim fix file still holding a lock on the given package.
3. If a `pre_remove` script is specified, it is run. If the `pre_remove` script returns a failure, the **emgr** command halts the remove operation.
4. **emgr** checks that the interim fix being removed is not a prerequisite for another installed interim fix.
5. The interim fix is removed. If the interim fix was installed with a standard installation operation, the **emgr** command replaces the current interim fix files with the previously saved files. If the installation was a mount installation operation, the **emgr** command unmounts the interim fix files and removes them from the system.
6. If a `post_remove` script is specified, it is run. If the `post_remove` script returns a failure, the **emgr** command halts the installation.
7. Reboot processing occurs. If the interim fix package specified that a reboot is required, the **emgr** command issues a message to the user and make any necessary changes to the boot image. The **emgr** command does not reboot the system automatically.

   Note: You can use **epkg** to specify the reboot scenario you want when you install another package. For more information on reboot scenarios, see “Interim fix user-specified package components” on page 369.
8. At this point, all removal steps have succeeded and the **emgr** command removes the remaining interim fix data from the database and save directories.

The summary and cleanup phase:

These steps occur during the summary and cleanup phase of removing an interim fix.

1. The **emgr** command issues a summary of all operations and results. If more than one interim fix package was specified with an input file using the **-f** flag, the **emgr** command reports for each interim fix package.
2. The **emgremgr** command cleans up any temporary directories and files. It also unloads any memory modules that have been loaded.

Listing interim fixes:

The **emgr** command lists data on installed interim fixes with various levels of verbosity.

The syntax for listing interim fixes is as follows:

```
emgr -l [-L Label | -n interim fixNumber | -u VUID ] [-v{1|2|3}] [-X] [-a path]
```

By default, the **emgr** command reports data on all installed interim fix. You can specify an individual interim fix by using one of the interim fix identification methods. For information about the interim fix identification methods, refer to "Referencing interim fixes" on page 363.

The default level of verbosity is 1. You can specify up to level 3 with the **-v** flag. The verbosity levels include the following information:

LEVEL 1

Lists one interim fix per line with the following information:

- Interim fix ID
- Interim fix state
- Install time
- Interim fix abstract

LEVEL 2

Lists the following information:

- All LEVEL 1 information
- Virtually unique ID (VUID) for each interim fix file
- Number of interim fix files
- Location for each interim fix file
- Package for each interim fix file
- Installer for each interim fix file
- Mount installation (`yes` or `no`) for each interim fix file

LEVEL 3

Lists the following information:

- All LEVEL 2 information
- Reboot requirement (`yes` or `no`)
- Prerequisite files needed
- Pre-install script
- Post-install script
- Pre-remove script
- Post-remove script
- File type for each interim fix file
- File size for each interim fix file
- Checksum for each interim fix file

- Access ownership and modes for each interim fix file
- Prerequisite information
- Interim fix description
- Archive member name for each interim fix file
- If this is a mount installation operation, then display the mount status for each interim fix file
- Reboot scenario for each interim fix file
- Interim fix to interim fix prerequisites for each interim fix file
- Packages that will be locked when the interim fix is installed
- Interim fixes that will be superseded when the interim fix is installed
- Authorized Program Analysis Report (APAR) information

Listing interim fix APAR information with the instfix command:

The **instfix** command can be used to list Authorized Program Analysis Report (APAR) information about fileset updates as well as interim fixes.

All of the **instfix** command functions are not available for the interim fixes. Only the **-f**, **-i**, **-k**, **-q**, **-r**, **-t**, and **-v** flags can be used. You cannot install the interim fixes by using the **instfix** command.

Some examples of use cases follow:

- To list APAR numbers that are associated with all types of fixes, run the following command:

  ```
  instfix -i
  ```

  Output:

  ```
  ...
  All filesets for IV14386 were found.
  All filesets for IV33073 were found.
  All filesets for IV25608 were found.
  Interim fix 'test' associated with IV12345 is installed.
  Interim fix 'test2' associated with IV25608 is installed.
  ```

- To list APAR numbers and abstracts associated with all types of fixes, run the following command:

  ```
  instfix -iv
  ```

  Output:

  ```
  ...
  IV19614 Abstract: AIX: Occassional missing FS info (incorrect mntctl use)
  Fileset rsct.core.fsrm:3.1.5.0 is applied on the system.
  Fileset rsct.opt.storagerm:3.1.5.0 is applied on the system.
  All filesets for IV19614 were found.
  IV12345 Abstract: Interim fix test
  Interim fix 'test' associated with IV12345 is installed.
  IV25608 Abstract: Interim fix test 2
  Interim fix 'test2' associated with IV25608 is installed.
  ```

- To limit the list to interim fixes, run the following command:

  ```
  instfix -it i
  ```

  Output:

  ```
  Interim fix 'test' associated with IV12345 is installed.
  Interim fix 'test2' associated with IV25608 is installed.
  ```

- To limit the list to interim fixes and include abstracts, run the following command:

  ```
  instfix -ivt i
  ```

  Output:

```
IV12345 Abstract: Interim fix test
Interim fix 'test' associated with IV12345 is installed.
IV25608 Abstract: Interim fix test 2
Interim fix 'test2' associated with IV25608 is installed.
```

- To query for a specific APAR number, run the following command:

  ```
  instfix -ik IV25608
  ```

  Output:

  ```
  All filesets for IV25608 were found.
  Interim fix 'test' associated with IV25608 is installed.
  ```

- To limit query to interim fixes, run the following command:

  ```
  instfix -ik IV25608 -t i
  ```

  Output:

  ```
  Interim fix 'test' associated with IV25608 is installed.
  ```

- To query for multiple APAR numbers, run the following command:

  ```
  instfix -ik "IV12345 IV25608"
  ```

  Output:

  ```
  Interim fix 'test' associated with IV12345 is installed.
  Interim fix 'test2' associated with IV25608 is installed.
  ```

Checking interim fixes:

The **emgr** command checks the status of installed interim fixes.

The syntax for interim fix checking is as follows:

```
emgr -c [-L Label | -n interim fixNumber | -u VUID | -f ListFile] [-w Directory] [-a path] \
[-v{1|2|3}] [-X]
```

By default the **emgr** command verifies all installed interim fixes. You can specify an individual interim fix by using one of the interim fix identification methods or specify several individual interim fixes by using a list file. For information about the interim fix identification methods, refer to "Referencing interim fixes" on page 363.

The default level of verification is 1. You can specify up to level 3 with the **-v** flag. The verification levels include the following checks:

LEVEL 1

Checks the following information:

- Interim fix data and state
- If this is a mount installation operation, then check the interim fix mount status for all files

  Note: If the interim fix file is unmounted, the **emgr** command changes the interim fix state to `UNMOUNTED`
- Interim fix checksum for all interim fix files or archive members

LEVEL 2

Checks the following information:

- All LEVEL 1 checks
- Interim fix ownership and mode for all interim fix files or archive members

LEVEL 3

Checks the following information:

- All LEVEL 2 checks

- All prerequisites
- All interim fix to interim fix prerequisites, including the following:
  - All interim fix prerequisites for the interim fix package being checked are installed.
  - All interim fix xrequisites for the interim fix package being checked are not installed.
  - There are no installed interim fixes that list the interim fix being checked as an xrequisite.

Performing an interim fix mount installation operation:

If the **-m** flag is specified during interim fix installation, the **emgr** command performs a mount installation operation of the interim fix package.

This means that the existing files that are being fixed are not removed from their present locations. Instead they are over-mounted by the interim fix files. This approach has both advantages and disadvantages. One advantage is that a system reboot unmounts all of the interim fixes. This means that any interim fix that created a serious problem is not mounted after a reboot. The disadvantages are that the administrator must monitor the mount status of interim fixes and some interim fixes may not be removed without a reboot.

The mount installation operation is not supported with interim fix packages that deliver new files.

The interim fix mount and unmount operation:

The **emgr** command mounts or unmounts interim fixes that have been installed using the mount installation operation.

The syntax for interim fix checking is as follows:

```
emgr -M | -U [-L Label | -n interim fixNumber | -u VUID | -f ListFile][-w Directory] [-a path] [-X]
```

By default, the **emgr** command applies the mount or unmount operation to all installed interim fixes. You can specify an individual interim fix by using one of the interim fix identification methods or specify several individual interim fixes by using a list file. For more information about the interim fix identification methods, see “Referencing interim fixes” on page 363.

Using the mount operation with the **-M** flag, the **emgr** command attempts to mount all interim fix files that are unmounted. If all interim fix files are successfully mounted, and the previous interim fix state was `UNMOUNTED`, then the **emgr** command changes the interim fix state to `MOUNTED`.

Using the unmount operation with the **-U** flag, the **emgr** command attempts to unmount all interim fix files that are mounted. If at least one interim fix file is successfully unmounted, and the previous interim fix state was `MOUNTED`, then the **emgr** command changes the interim fix state to `UNMOUNTED`.

Using the interim fix display package locks operation:

The **display package locks** operation displays all packages that are locked by interim fix manager, their installer, and the locking label or labels.

The syntax for the **display package locks** operation is as follows:

```
emgr -P [Package] [-a path] [-X]
```

By default, the **emgr** command lists all locked packages. The user can specify an individual package as an argument to the **-P** flag.

Using the interim fix force removal operation:

The **force removal** operation removes interim fix data.

This operation also unlocks all interim fix packages associated with the interim fix label without removing the actual interim fix files, executing any removal scripts, or boot processing. The force removal operation can only be run on one interim fix at a time, and the interim fix label is required to identify the target interim fix. The syntax for performing a force removal operation is as follows:

```
emgr -R interim fix fixLabel [-w Directory] [-a path] [-X]
```

Note: The force removal operation must be considered as an emergency procedure. It must *only* be run if all other methods to remove the interim fix have failed. This method can create inconsistencies on the target system.

Additional interim fix information

The following are links to additional fix information.

Generating and using the MD5 checksum:

At the beginning of any operation involving **epkg** formatted images, **emgr** looks on the system for a supported command that generates an MD5 checksum. If a command is located, **emgr** executes this command and displays the resulting MD5 checksum.

You can then cross check this MD5 checksum with a secured source. If a command is not located, **emgr** takes no further action. You can force set an explicit path to a command that generates a checksum by exporting the **EMGR_MD5_CMD** shell variable. This variable should contain the absolute path to the command. **emgr** does not verify that the user set command in the **EMGR_MD5_CMD** variable is an actual command that generates an MD5 checksum. The syntax used by **emgr** to generate the MD5 checksum is as follows:

```
$EMGR_MD5_CMD epkg image file
```

The expected output is the MD5 checksum as the first word in the output.

Note: This feature is not supported in the original release of interim fix management. You should update to the latest level of interim fix management to enable this feature. To update interim fix management, update `bos.rte.install` to the latest level.

Referencing interim fixes:

The following are methods for referencing interim fixes.

Reference by Label

Each interim fix that is installed on a given system has its a unique interim fix label. This label is called the *unique key* that binds the different database objects. To reference an interim fix by label, pass the label as an argument to the **-L** flag.

For example, to run a check operation on an interim fix with label `ABC123`, type the following:

```
# emgr -cL ABC123
```

Reference by Interim Fix ID

Each interim fix that is installed on a given system has a unique interim fix ID. This ID is the order number in which the interim fix is listed in the interim fix database. Using this option may be convenient if performing operations on interim fixes based on interim fix listings. The **emgr** command converts the interim fix ID into an interim fix label before performing the given operation. To reference an interim fix by ID, pass the ID as an argument to the **-n** flag.

For example, to run a check operation on the first interim fix with an ID equal to 1, type the following:

```
# emgr -cn1
```

Note: Emergency fix IDs are valid for short periods of time and change as interim fixes are removed and added. *Always* verify the current interim fix ID number by listing the interim fix using the **-l** flag.

Reference by VUID

The VUID is used to differentiate packages that have the same label. Unlike Authorized Program Analysis Reports (APARs), which are officially tracked, emergency fixes are not tracked by any organization, so it is possible to have two interim fix packages with the same label. However, the **emgr** command does not allow the installation of more than one interim fix with the same label. The **emgr** command converts the VUID into an interim fix label before performing the given operation. For example, to list an installed interim fix with VUID equal to `000775364C00020316020703`, type the following:

```
# emgr -l -u 000775364C00020316020703
```

The VUID is also displayed in the preview section of the interim fix installation and removal operations, and when using the listing operation with verbosity level 2 or greater. For more information on verbosity levels, see "Listing interim fixes" on page 359

Generating interim fix list files:

You can perform operations on a set of interim fixes by specifying a list file.

For the installation operation, the list file must contain one interim fix package location per line. For the removal operation and the mount and unmount operations, the list file must have one interim fix label name per line. You can use the **-f** flag on the **emgr** command to specify a file that contains one of the following:

- A list of package locations for the installation operation (one per line)
- A list of interim fix labels for the remove, mount, unmount, and check operations (one per line)

The **emgr** command ignores any blank lines, or lines where the first non-white space character is the # character.

Understanding interim fix states:

The **emgr** command maintains a state for each installed interim fix.

The following installed interim fix states are maintained by the **emgr** command:

S=STABLE

The interim fix was installed with a standard installation (-e flag), and successfully completed the last installation operation. To verify the interim fix details, run a check operation on the given interim fix or interim fixes.

M=MOUNTED

The interim fix was installed with a mount installation operation, and successfully completed the last installation or mount operation. A state of `MOUNTED` does not mean all interim fixes are currently mounted. For example, the interim fixes might have been manually unmounted. This state represents the **emgr** command's previous actions and determination of the mount status. To verify the interim fix details, including mount status, run a check operation on the given interim fix or interim fixes.

U=UNMOUNTED

The interim fix was installed with a mount installation operation and one or more interim fix files were unmounted in a previous **emgr** command operation. The state of `UNMOUNTED` does not mean that all interim fixes are currently unmounted. For example, the interim fixes might have been manually mounted or partially mounted. This state represents the **emgr** command's previous actions and determination of the mount status. To verify the interim fix details, including mount status, run a check operation on the given interim fix or interim fixes.

B=BROKEN
An unrecoverable error occurred during an installation or removal operation. The status of the interim fix is unreliable. You can attempt to remove this interim fix and reinstall it from the interim fix package.

I=INSTALLING
The interim fix is in the process of installing. Normally, this state occurs only for a brief time during interim fix installation. However, if an interim fix installation is suddenly interrupted (such as in a sudden power loss or a system crash), and the **emgr** command is unable to clean up the failed installation, an interim fix might be left in the `INSTALLING` state. You can attempt to remove this interim fix and reinstall it from the interim fix package.

Q=REBOOT REQUIRED
The interim fix was installed successfully and requires a reboot to fully integrate into the target system. After you reboot the target system, **emgr** changes the interim fix state to `STABLE`.

Note:

1. This feature is not supported in the original release of interim fix management. You should update to the latest level of interim fix management to enable this feature. To update interim fix management, update `bos.rte.install` to the latest level.
2. **emgr** is called by **init** with the **-B** bootup flag during system start. **emgr** examines the state data for all interim fixes and changes the interim fix states as necessary. The `/etc/inittab` entry `rcemgr` is created the first time **emgr** installs an interim fix file. `rcemgr` is set to not block or respawn. All `rcemgr` activities and output are logged to the **emgr** log in `/var/adm/ras/emgr.log`. Do not directly execute **emgr** with the **-B** flag.

R=REMOVING
The interim fix is in the process of being removed. Normally, this state occurs only for a brief time during interim fix removal. However, if an interim fix installation is suddenly interrupted (such as in a sudden power loss or a system crash), and the **emgr** command is unable to clean up the failed installation, an interim fix might be left in the `REMOVING` state. You can attempt to remove this interim fix and reinstall it from the interim fix package.

T = TESTED
Specifies that the interim fix was tested. Reflects the value of the **epkg** command -T flag that may be used during package creation.

P = PATCHED
Concurrent update has been patched directly in memory. Corresponding binaries on disk have not been modified.

N = NOT PATCHED
Concurrent update has been "updated by" a follow-on concurrent update, making the patch inactive.

State is also set during reboot to change the state of in memory only concurrent updates that were in the PATCHED state.

SP = STABLE + PATCHED
Concurrent update has been patched in memory and the corresponding on-disk binaries have been replaced. The fix will now persist on the system across reboots.

SN = STABLE + NOT PATCHED
A concurrent update in the STABLE + PATCHED state will be moved to this state when it is "updated by" a follow-on concurrent update, making the patch inactive.

State may also be set if an interim fix containing a concurrent update in the NOT PATCHED state is committed to disk.

QP = BOOT IMAGE MODIFIED + PATCHED
Same as Stable + Patched except that, in replacing on-disk binaries, a file belonging in the boot image was modified and bosboot was invoked.

QN = BOOT IMAGE MODIFIED + NOT PATCHED
A concurrent update in the BOOT IMAGE MODIFIED + PATCHED state will be moved to this state when it is "updated by" a follow-on concurrent update, making the patch inactive.

State may also be set if an interim fix containing a concurrent update in the NOT PATCHED state is committed to disk.

RQ = REMOVING + REBOOT REQUIRED
When an interim fix that was committed to disk has been removed, the system must be rebooted to remove all patched code from memory. The interim fix will be removed from the Interim Fix database by the **rc_emgr** function called by init at boot time.

Logging interim fixes:

These operations are logged in the `/var/adm/ras/emgr.log` **emgr** log file.
- Installation
- Removal
- Checking
- Mounting
- Unmounting
- Force Removal

Cleaning up interim fix installation failures:

The failure-cleanup procedure is run when an interim fix installation operation fails after the installation preview (and **pre_install** script, if specified).

The failure-cleanup procedure attempts to reverse any of the changes that have already been made by the installation process and is similar to the removal phase of the interim fix removal operation. This procedure sets the **EMGR_UNDO** global environment variable to 1 and allows packaging to take different paths in the **pre_remove** and **post_remove** scripts.

Managing interim fix files when using the Trusted Computing Base:

The **emgr** command automatically detects if a system is enabled with the Trusted Computing Base (TCB).

If TCB is enabled, the **emgr** command registers all of the installed interim fixes with the interim fix database. When the interim fixes are removed, the **emgr** command restores the original TCB data. Because mount installation operations can create variations in file attributes when interim fix files are mounted and unmounted, mount installation operations are not supported on a TCB-enabled system and are blocked by the **emgr** command.

If you do not want the **emgr** command to automatically manage TCB data, export the **EMGR_IGNORE_TCB** variable and set this variable to any value that is not null. When the **EMGR_IGNORE_TCB** variable is set, the **emgr** command behaves as if the system is not TCB-enabled. If the **EMGR_IGNORE_TCB** variable is set on a TCB-enabled system, you might be required to manually manage interim fix files within TCB.

To check if TCB is enabled on your system, run the **/usr/bin/tcbck** command. If a usage statement is returned, TCB is enabled. Otherwise, a message indicating that TCB is not enabled is returned.

Using emgr to manage interim fix command paths.:

The **emgr** command calls one or more of the following UNIX commands.

ar
awk
cat
chmod
chown
compress
cp
date
df
diff
du
egrep
fuser
id
ksh
ln
ls
mkdir
mount
mv
printf
ps
rm
rmdir
sed
sleep
sort
sum
tail
tar
tee
touch
umount
uname
vi
wc
zcat

The **emgr** command calls one or more of the following AIX commands:

aclget
aclput
bosboot
lslpp

odmchange
odmget
slibclean
tcbck

The **emgr** command looks for the UNIX and AIX commands previously listed in the following path order:

1. /usr/emgrdata/bin
2. /usr/bin
3. /usr/sbin
4. /bin
5. /sbin
6. /usr/local/bin
7. /usr/local/sbin

The `/usr/emgrdata/bin` directory is a secured directory that is created the first time the **emgr** command is run.

If you are attempting to install or remove an interim fix for one of the commands that the **emgr** command uses, you might not be able to successfully complete the operation. To solve this problem, do the following:

1. Manually install the interim fix file into the `/usr/emgrdata/bin` directory.
2. Perform the **emgr** operation.
3. Remove the manually installed interim fix file from the `/usr/emgrdata/bin` directory.

Using this method, the interim fix is registered and tracked with interim fix manager and all other **emgr** command processing takes place.

If the interim fix file is the `/usr/bin/ksh` file and the problem it fixes prevents the **emgr** command's operations from succeeding, then do the following:

1. Back up the original `/usr/bin/ksh` file.
2. Manually install the `/usr/bin/ksh` interim fix file to `/usr/bin/ksh`.
3. Perform the **emgr** command installation or remove operation.

Understanding interim fix integration with installp update images.:

The interim fix management commands use an APAR reference file to associate interim fixes with APAR numbers.

When the APAR images are available, the **installp** command matches the APAR numbers contained in the update image with the APAR numbers installed with the interim fix. If all APAR numbers are matched by the update, the interim fixes will automatically be removed.

Creating interim fix packages

If you need to create your own interim fix and package it for distribution, use the **epkg** command to package the interim fix.

The **epkg** command can be run in two modes: *interactive* and *template-based*. The interactive method prompts the user with several questions and constructs the interim fix package based on the answers. The template-based method uses an interim fix control file that is pre-filled with default answers that are then asked in interactive mode. The interim fix package can then be installed by the **emgr** command.

By using an interim fix control file as a template, interim fix packages can be created noninteractively. For an example of a completed interim fix control file, see the **epkg** command.

Interim fix user-specified package components:

The listed interim fix control-file components are part of the overall interim fix package and are not related to specific files.

ABSTRACT
Describes the interim fix package. The abstract is limited to 38 bytes.

DESCRIPTION
Contains a detailed description of the interim fix package that is being installed.

APARREF
Specifies the location of a file that contains the APAR number or numbers associated with this interim fix. This component is required. The file must be contain one APAR number per line.

E2E_PREREQ
Lists the interim fix label names of interim fixes that are prerequisites to the interim fix package being installed. Using this file causes **emgr** to check if the interim fix **PREREQ** label is installed. If the prerequisite is not installed, **emgr** aborts installation of the interim fix package. You can also use this file to specify an **XREQ** interim fix label. Specifiying **XREQ** interim fix labels causes **emgr** to not install the interim fix package if the specified interim fix is installed. The maximum number of supported interim fix labels is 32. You can specify the interim fix labels to check for in the following ways.

- Specify the file location with the **-g** flag. For example, to specify interim fix **prereq.epkg**, type the following:

  ```
  # epkg -g /tmp/efixprereq.epkg myefix
  ```

- Use the **-v** flag in interactive mode for extended options, and type the file location when prompted by **epkg**. For example, to specify **interim_fixprereq.epkg**, when prompted, type the following:

  ```
  Enter the location for the supersede file or "." to skip.
     -> /tmp/interim_fixprereq.epkg
  ```

- Set the **E2E_PREREQ** attribute in the interim fix control file to the local file location of the interim fix prerequisite file. For example, to specify **interim_fixprereq.epkg**, set the attribute as follows:

  ```
  E2E_PREREQ=/tmp/interim_fixprereq.epkg
  ```

The format of the interim fix prerequisite file entries is as follows (where **RequisiteType** is *PREREQ* or *XREQ*):

```
EfixLabel RequisiteType
```

Comments beginning with a "#" sign and leading white space are ignored. For example:

```
oldefix1 PREREQ  # Make sure oldefix1 is already installed
oldefix4 XREQ    # Make sure oldefix4 is NOT installed
```

Note: This feature is not supported in the original release of interim fix management. You should update to the latest level of interim fix management to enable this feature. To update interim fix management, update **bos.rte.install** to the latest level.

PKGLOCKS
Lists the packages that should be locked by **emgr** in addition to those that are automatically locked based on file ownership. You should specify the name of the package, the package lock action (either *ALWAYS* or *IFINST*), and the package file type. *ALWAYS* means always attempt to lock this package, and a failure to lock the package results in interim fix installation failure. *IFINST* means attempt to lock this package only if the package is installed, and failure to lock an

installed package results in interim fix installation failure. The maximum number of supported interim fix labels is 32. You can specify the packages to be locked in the following ways.

- Specify the file location with the **-l** flag. For example, to specify **pkglock.epkg**, type the following:

  ```
  # epkg -l /tmp/pkglock.epkg myefix
  ```

- Use the **-v** flag in interactive mode for extended options, and type the file location when prompted by **epkg**. For example, to specify **pkglock.epkg**, when prompted, type the following:

  ```
  Enter the location for the supersede file or "." to skip.
     -> /tmp/pkglock.epkg
  ```

- Set the **PKGLOCKS** attribute in the interim fix control file to the local file location of the package to be locked. For example, to specify **pkglock.epkg**, set the attribute as follows:

  ```
  PKGLOCKS=/tmp/pkglock.epkg
  ```

The format of the interim fix package locks file entries is as follows:

```
PackageName PackageAction PackageType
```

Comments beginning with a "#" sign and leading white space are ignored. In the following example, **emgr** will always attempt to lock **bos.rte.lvm** during installation and will unlock it on removal. **emgr** will lock **bos.games** if (and only if) it is installed and will unlock it on removal (if locked).

```
bos.rte.lvm ALWAYS installp
bos.games   IFINST installp
```

Note: This feature is not supported in the original release of interim fix management. You should update to the latest level of interim fix management to enable this feature. To update interim fix management, update **bos.rte.install** to the latest level.

PRE_INSTALL
Runs after an installation preview and before any interim fix files are installed. Failure in the `pre_install` script causes the interim fix package installation to be aborted. This script is useful for doing any preinstallation checking or work. Because the **emgr** command does not call a failure-cleanup procedure for preinstallation failures, this script performs failure cleanup (related to the script) before it exits. This component is optional.

POST_INSTALL
Runs after all interim fix files have been successfully installed. A failure in the `post_install` script causes the installation to fail and causes interim fix manager to run a failure-cleanup procedure. This component is optional. For more information about the `post_install` script, refer to “Installing and managing interim fix packages” on page 354.

PRE_REMOVE
Runs after the removal preview and before any interim fix files are removed during a remove operation and in the first stage of a failure-cleanup procedure. A failure in the `pre_remove` script causes the given operation to fail. In the case of a failure-cleanup procedure, the **emgr** command sets an **EMGR_UNDO** global environment variable to 1. If necessary, the **EMGR_UNDO** variable is used to take different actions for removal as opposed to a failure-cleanup. This component is optional.

POST_REMOVE
Runs after interim fix files are removed during a remove operation and a failure-cleanup procedure. A failure in the post-remove script causes the given operation to fail. In the case of a failure-cleanup procedure, the **emgr** command sets an **EMGR_UNDO** global environment variable to 1. The **EMGR_UNDO** variable is used to take different actions for removal as opposed to a failure-cleanup (if necessary). This component is optional.

REBOOT
Indicates whether a reboot operation is required for this interim fix. You can use this variable to specify one of the following reboot scenarios.

- Reboot is not required.
- Reboot is required, and the boot image will be rebuilt.
- Reboot is required, and the boot image will not be rebuilt.

You can specify the which of these reboot scenarios you want in the following ways.

- Specify the reboot scenario with the **-r** flag. Arguments for this flag are *n* (reboot is not required), *y* (reboot required and the boot image will be rebuilt), and *o* (reboot is required, but the boot image will not be rebuilt). For example, the following command specifies that a reboot is not required:

  ```
  # epkg -r n
  ```

- Use the **-v** flag in interactive mode for extended options, and select the reboot scenario you want when prompted by **epkg**. For example:

  ```
  Select reboot policy for this efix package:
     1) Reboot is NOT required.
     2) Reboot is required. The boot image will be rebuilt.
     3) Reboot is required. The boot image will NOT be rebuilt.
  ```

- Set the **REBOOT** and **BUILD_BOOT_IMAGE** attributes in the interim fix control file to the appropriate values for the reboot scenario you want. For example, to specify that a reboot is not required, set the attributes as follows:

  ```
  REBOOT=no
  BUILD_BOOT_IMAGE=no
  ```

  To specify that a reboot is required and the boot image will be rebuilt, set the attributes as follows:

  ```
  REBOOT=yes
  BUILD_BOOT_IMAGE=yes
  ```

  To specify that a reboot is required and the boot image will not be rebuilt, set the attributes as follows:

  ```
  REBOOT=yes
  BUILD_BOOT_IMAGE=no
  ```

  Note:

  1. This feature is not supported in the original release of interim fix management. You should update to the latest level of interim fix management to enable this feature. To update interim fix management, update **bos.rte.install** to the latest level.

PREREQ

Contains **installp** prerequisites. This component is optional.

- The file has one prerequisite per line.
- The format of the prerequisite entry is as follows:

  Fileset Min Level Max Level Type

  Fileset The name of the requisite `installp` fileset.

  Min Level
  The minimum level for the requisite fileset. The specification of `NONE` indicates no minimum level.

  Max Level
  The maximum level for the requisite fileset. The specification of `NONE` indicates no maximum level.

  Type The following types are supported: `PREREQ` and `IFREQ`. `PREREQ` is the default type and requires that the requisite fileset meets all criteria. `IFREQ` requires that requisite fileset meet all criteria only if it is installed.

- Blank lines or lines that start with # are ignored, as shown in the following examples:

```
# Require that abc.xyz is installed at any level:
  abc.xyz NONE NONE
# Require that bos.rte.lvm is installed at level 5.1.0.10 or above:
  bos.rte.lvm 5.1.0.10 NONE
# Require bos.mp be between 5.1.0.0 and 5.1.0.40 if it is installed:
  bos.mp 5.1.0.0 5.1.0.40 IFREQ
```

SUPERSEDE

Specifies the interim fix label name of an interim fix or interim fixes that are to be superseded when an **epkg** is installed. Using this file causes **emgr** to remove any interim fix labels that are specified in this file (if they are installed) before installing the interim fix package. Failure to remove an installed superseded interim fix aborts the installation of the interim fix package. The maximum supported number of superseded labels is 32. You can specify the superseded file in the following ways.

- Specify the file location with the **-S** flag. For example, to specify **susperseded.epkg**, type the following:

  ```
  # epkg -S /tmp/superseded.epkg myefix
  ```

- Use the **-v** flag in interactive mode for extended options, and type the file location when prompted by **epkg**. For example, to specify **susperseded.epkg**, when prompted, type the following:

  ```
  Enter the location for the supersede file or "." to skip.
     -> /tmp/superseded.epkg
  ```

- Set the **SUPERSEDE** attribute in the interim fix control file to the local file location of the superseded file. For example, to specify **susperseded.epkg**, set the attribute as follows:

  ```
  SUPERSEDE=/tmp/superseded.epkg
  ```

The format for the list of superseded files is one interim fix label to be superseded per line. Comments beginning with a "#" sign and leading white space are ignored. For example:

```
# Requisites for efix myefix3
myefix1
myefix2
```

Note: This feature is not supported in the original release of interim fix management. You should update to the latest level of interim fix management to enable this feature. To update interim fix management, update **bos.rte.install** to the latest level.

Interim fix file components:

The following interim fix control-file components are related to specific files. The maximum number of interim fix files for each interim fix that the **epkg** and **emgr** commands support is 200.

EFIX_FILE_NUM

Number of the given file (1 - 200).

SHIP_FILE

Local file location that the **epkg** command is archiving into the interim fix package. Specify either an absolute or relative path to this file. The ship file is the interim fix that is delivered.

TARGET_FILE

Target file location where the **SHIP_FILE** is installed. This location is located on the system where the interim fix package is installed. Specify the absolute path to this file. If this file is part of a registered package, such as an RPM or **installp** package, you must specify the tracked location.

INSTALLER

This variable represents the type of installer that owns the interim fix package. Valid integer choices are as follows:

1 Tracked by **installp**

2 Tracked by RPM

3 Tracked by **ISMP**

4 Tracked by another installer

5 New file that will be tracked by **installp**

6 New file that will be tracked by RPM

7 New file that will be tracked by **ISMP**

8 New file that will be tracked by another installer

9 Not tracked by any installer

TYPE This is the type of file that is being installed. The valid choices are as follows:

1 Standard file or executable file

2 Library or archive member

An example of **TYPE 1** is the `/usr/bin/ls` file or the `/usr/bin/rm` file. An example of **TYPE 2** is the **shr.o** archive member as a member of the **libc.a** library.

ACL Specifies the access attributes (mode and ownership) for a given file. If this attribute is set to `DEFAULT`, the **emgr** command maintains the current permissions of the file to be replaced. However, if the target file is a new file or the user wants to specify permissions using the **-v** flag, the **ACL** attribute can be entered with the *Owner:Group:OctalModes* syntax, similar to the following:

```
ACL= root:system:555
```

AR_MEM
Specifies the name of the archive member. This option is only valid if `TYPE=2`. In this case, **SHIP_FILE** represents the local location of the archive member that is being shipped, **TARGET_FILE** represents the target archive, and **ACL** applies to the archive member. For example, the following attributes ship the **myshr.o** local file to the **shr.o** member in the `/usr/ccs/lib/libc.a` target archive:

```
TYPE=2
SHIP_FILE=/home/myshr.o
TARGET_FILE=/usr/ccs/lib/libc.a
AR_MEM=shr.o
```

Interim fix automatic common components:

The listed components are part of the overall interim fix package and are not related to specific files.

These components are automatically determined by the **epkg** command. Typically, the user does not set the following components:

DATE Date and time that the backup was made.

INSTWORK
Amount of space (in 512 byte-blocks) required for unpacking the interim fix metadata.

VUID Virtually Unique ID. A combination of time and **cpuid**, this ID can be used to differentiate fixes that are otherwise identical.

QNEXT and QORDER
Internal trackers for interactive mode. If you are using an interim fix control file in nonprompted mode, make sure **QNEXT** and **QORDER** are not set, or set to `QEND`.

Interim fix automatic file components:

The listed components are related to specific files. These components are automatically determined by the **epkg** command. Typically, the user does not set these components.

CKSUM
File checksum for the given file

SIZE Size for the given file

Running the epkg command in interim fix interactive mode:

By default, the **epkg** command is run in interactive mode. The only required parameter is the interim fix label.

If you interrupt a running **epkg** command session, the interim fix control file is saved. If you start a new session with the same interim fix label, you are asked whether you want to keep working with the previous interim fix control file. You can use the **-u** flag to answer this question.

The **epkg** command maintains a record of the question order and allows the user to navigate between questions by using subcommands. Also, the **epkg** command remembers the previous answer the user provided and sets that answer as the default answer. The **epkg** subcommands are the following:

b! Returns to the previous question.

s! Shows the current interim fix control file.

q! Quits without saving interim fix control file (using the Ctrl-C key sequence will prompt this subcommand).

h! Displays help information for the current question.

The **epkg** command asks the following questions, one at a time:

```
1. Enter interim fix abstract [38 bytes maximum]:
   ** If "-s" flag is specified, go to question #3 ***

2. Does this interim fix deliver one or more files ? (yes/no):
   ** If "no", go to question #9 **

3. Enter the local ship file location for interim fix file number 1:

4. Enter target location for interim fix file number 1:

5. Select file type for interim fix file number 1:
   1) Standard (file or executable)
   2) Library member

6. Select the installer which tracks the file that is being fixed by interim fix
   file number 1:
   1) Currently tracked by installp.
   2) Currently tracked by RPM.
   3) Currently tracked by ISMP.
   4) Currently tracked by another installer.
   5) This is a NEW file that will be tracked by installp.
   6) This is a NEW file that will be tracked by RPM.
   7) This is a NEW file that will be tracked by ISMP.
   8) This is a NEW file that will be tracked by another installer.
   9) Not tracked by any installer.

   *** If "-m flag" and not new go to #7.1 ***
   *** If new, go to #7.2 ***
   *** Else, go to #8 ***

7.1 Enter the ACL for file 1 in the format of <owner>:<group>:<octal modes>.
For example to make the user="root", the group="system", and the modes "444",
you would enter root:system:444. Enter "." if you want to keep the default
(i.e. current) permissions on the existing target file.
```

```
7.2 Enter the ACL for file 1 in the format of <owner>:<group>:<octal modes>.
For example to make the user="root", the group="system", and the modes "444",
you would enter root:system:444.

8. Are there more interim fix files ? (yes/no):
   ** If "yes", increment file and go to question #3 **
   ** If "no" and "-s" flag, go to #14 **
   ** If "no" go to question #9 **

9.  Enter the local location for the pre-install script or "." to skip.

10. Enter the local location for the post-install script or "." to skip.

11. Enter the local location for the pre-remove script or "." to skip.

12. Enter the local location for the post-remove script or "." to skip.

14. Is a reboot required after installing this interim fix ? (yes/no):

15. Enter the location for the APAR reference file.

16. Enter the local location for the installp prerequisite file or "." to skip.
    *** This question is skipped if -p flag ***

17. Enter the local location for the interim fix description file or "." to compose it
in an editor:
     *** This question is skipped if "-d" flag is specified           ***
     *** If the description file is not specified, the user will be   ***
     *** put into an editor to compose it. The user can specify       ***
     *** which editor to use by setting the EDITOR global environment ***
     *** variable. The default editor is /usr/bin/vi.                 ***
```

After all of the questions are answered, the **epkg** command verifies the interim fix control file and creates a compressed **tar** package that can be installed by using the **emgr** command.

Interim fix installation and the Live Update function

Interim fixes that contain kernel extensions or a new kernel and that requires the system to be restarted can now be installed by using the AIX Live Update function if the interim fixes are marked as `LU CAPABLE`. Install the interim fix in a preview mode by using the **emgr -p -e** *ifix_pkg* command, and search for `LU CAPABLE` occurrence in the output to determine if the interim fix is suitable for the Live Update function.

This option is available when you use the **geninstall** command with the **-k** flag to install the interim fix. This option is also available in the following SMIT menus:

Install Software
The `smitty install_latest` fast path.

Install Software Bundle
The `smitty install_bundle` fast path.

Install and Update from ALL Available Software
The `smitty install_all` fast path.

The `bos.liveupdate.rte` fileset must be installed on the AIX operating system if you want to use the Live Update function.

Related concepts:

"Live Update"
Starting with AIX Version 7.2, the AIX operating system provides the AIX Live Update function that eliminates the workload downtime that is associated with AIX system restart that is required by previous AIX releases when fixes to the AIX kernel are deployed. The workloads on the system are not stopped in a Live Update operation, yet the workloads can use the interim fixes after the Live Update operation.

Live Update

Starting with AIX Version 7.2, the AIX operating system provides the AIX Live Update function that eliminates the workload downtime that is associated with AIX system restart that is required by previous AIX releases when fixes to the AIX kernel are deployed. The workloads on the system are not stopped in a Live Update operation, yet the workloads can use the interim fixes after the Live Update operation.

IBM delivers kernel fixes in the form of interim fixes to resolve issues that are reported by customers. If a fix changes the AIX kernel or loaded kernel extensions that cannot be unloaded, the host logical partition (LPAR) must be restarted. To address this issue, AIX Version 7.1, and earlier, provided concurrent update-enabled interim fixes that allow deployment of some limited kernel fixes to a running LPAR. All fixes cannot be delivered as concurrent update-enabled interim fixes. Starting with AIX Version 7.2, you can use the Live Update function to eliminate downtime that is associated with the AIX kernel update operation. This solution is not constrained by the same limitations as in the case of concurrent update enabled interim fixes.

AIX Version 7.2 Service Pack 1 contains significant fixes for AIX Live Update. You can download AIX Version 7.2 Service Pack 1 at the Fix Central website.

In AIX Version 7.2 with the 7200-01 Technology Level, or later, you can use the Live Update function to update service packs and technology levels for the AIX operating system.

In AIX Version 7.2 with the 7200-02 Technology Level, or later, partitions that are managed by IBM Power Virtualization Center (PowerVC) can use the Live Update function.

The Live Update concepts

In the AIX Live Update function, the logical partition (LPAR) where the operation is started is called the *original* partition. The operation involves another LPAR that is called the *surrogate* partition. *Checkpointing* a workload means freezing a running process and saving its current state. Checkpointing processes on an LPAR and restarting them later on another LPAR is called *mobility*.

If you plan to install updates by using the Live Update function, before you begin the installation, you must back up your system so that you can return to the previous operating level, if necessary, by restoring the system from the backup or by restarting your system from an alternate disk copy. The updates that are installed by using the Live Update function are always committed. Therefore, you cannot reject the updates later.

The updates for a service pack, technology level, and interim fixes are applied before starting the surrogate partition, and the running workloads are transferred from the original partition to the surrogate partition. The Live Update process involves the following steps:

1. If updates to a service pack or technology level are specified to be installed by using the Live Update function, the updates are applied and committed first on the original partition.
2. If any interim fixes are specified along with the service pack and technology level updates, the interim fixes are installed on the original partition.
3. The root volume group of the original partition (`orig-rootvg`) is cloned.
4. If only interim fixes are specified for the Live Update operation, the interim fixes are applied on the cloned volume group that serves as the boot volume group for the surrogate partition (`surr-boot-rootvg`).

5. After the surrogate partition is started and while the workloads are still running on the original partition, the root volume group of the surrogate partition is mirrored (`surr-mir-rootvg`).
6. The workload processes are check pointed and moved to the surrogate partition.
7. Workloads resume on the surrogate partition in a `chrooted` environment (changed root directory) on the original root volume group (`orig-rootvg`). During this process, the workloads continue to run without being stopped, although a short blackout time occurs when these workloads are suspended.
8. If the Live Update operation fails after step 1 and step 2, the updates and interim fixes installed on the system in these steps are not uninstalled. If the cause of the Live Update failure is corrected, you can attempt the Live Update operation again instead of restarting the original LPAR. In this scenario, updates or interim fixes are not specified for the Live Update operation because the updates are already installed.

The Live Update feature is intended for applying interim fixes that contain kernel changes or kernel extension changes that require a reboot. The interim fix might contain other files (for example, commands and libraries), and the Live Update feature does not change anything about the way these files are applied. For example, a shared library will be modified on the file system, but any running processes continues to use the old version of the library. Therefore, applications that require a library fix must be stopped and restarted to load the new version of the library after the fix is applied. In AIX® Version 7.2 with the 7200-01 Technology Level, or later, you can use the **genld -u** command to list the processes that are using the old version of any shared libraries or other objects that are updated. You can use the list that is displayed from the **genld -u** command to identify the processes that must be stopped and restarted to load the updated objects.

The Live Update operation is not a stand-alone command. It can be launched only through the **geninstall -k** option or Network Installation Manager (NIM). The inputs to the Live Update operation are supplied through the stanzas in the `/var/adm/ras/liveupdate/lvupdate.data` file. A template of this file is supplied with the system. You must edit this file to reflect your own configuration. The **geninstall** command uses a lock file, `/usr/lpp/.genlib.lock.check`, to guarantee that no other installation process can run simultaneously. A special line `INU_LKU_LOCK` in this lock file is used to indicate that other installations must be blocked. In another scenario, NIM can be used with the **-o cust** option from a centralized server to invoke the **geninstall** command on a target machine. In this case, the `/var/adm/ras/liveupdate/lvupdate.data` file is exported by the NIM master and mounted by the NIM client on the target machine.

The Live Update operation runs in one of the following modes:

Preview mode
: In preview mode, estimation of the total operation time, estimation of application blackout time, and estimation of resources such as storage and memory are provided to the user. These estimations are based on the assumption that the surrogate partition has the same resources in terms of CPU, memory, and storage as the original partition. All the provided inputs are validated and the Live Update limitations are checked.

Automated mode
: In automated mode, a surrogate partition with the same capacity as the original partition is created, and the original partition is turned off and discarded after the Live Update operation completes.

The mirror copy of the original root volume group (rootvg) is retained after the Live Update operation is complete. Thus, if you have installed only interim fixes with the Live Update function and if you want to return to the state of the system before you applied the interim fixes, the LPAR can be restarted from the disk that was specified as the mirror volume group (mirrorvg).

Alternatively, you can choose to install any updates or interim fixes on the original LPAR by using any installation method that is supported by the AIX® operating system. After these updates or fixes are

installed, you can use the Live Update function to load the updated kernel software without restarting the system. The Live Update process for this scenario involves the following steps:

1. Back up the system by using your preferred backup method. A backup is required if you want to restore the system to its previous state before the updates or interim fixes were installed.
2. Install the updates and interim fixes by using any supported installation method (Network Installation Manager (NIM) or installp).
3. If you must restart the system to apply the updates or interim fixes, you can use the Live Update function instead of restarting the system. The Live Update operation starts either through the **geninstall** command or NIM. The Live Update operation does not require you to specify any updates or interim fixes because the updates are installed on the system.
4. The root volume group of the original partition (`orig-rootvg`) is cloned.
5. After the surrogate partition is started and while the workloads are still running on the original partition, the root volume group of the surrogate partition is mirrored (`surr-mir-rootvg`).
6. The workload processes are check pointed and moved to the surrogate partition.
7. Workloads resume on the surrogate partition in a `chrooted` environment (changed root directory) on the original root volume group (`orig-rootvg`). During this process, the workloads continue to run without being stopped, although a short blackout time occurs when the workloads are suspended.
8. If the Live Update operation fails, correct the cause of the failure, and retry the process starting at step 3.

Related information:

geninstall Command

Planning for the Live Update operation

The AIX Live Update operation is an alternative method to apply an update.

To use the Live Update function, consider the following additional configuration steps:

1. Verify that the environment meets the requirements for the Live Update operation. For more information about the Live Update limitations, see "LPAR requirements for Live Update."
2. Create the `lvupdate.data` file. For more information about this file, see "Configuring resources for Live Update" on page 384.
3. Perform a Live Update operation either through Network Installation Manager (NIM) or by using the **geninstall** command. For more information about these procedures, see "Performing the Live Update operation by using NIM" on page 389 and "Performing the Live Update operation by using the geninstall command" on page 390.

LPAR requirements for Live Update:

Consider the following requirements for a logical partition (LPAR) to support the AIX Live Update feature:

- All I/O must be virtualized through the Virtual I/O Server (VIOS). The VIOS itself does not support the Live Update function.
- All the mounted file systems must be Enhanced Journaled File System (JFS2) or network file system (NFS). The CacheFS, Automount File System (AutoFS), or Autonomic Health Advisor File System (AHAFS) mounts must not be active.
- The LPAR can be managed by either Hardware Management Console (HMC) or IBM Power Virtualization Center (PowerVC):

  HMC-based Live Update operation

  If the LPAR is managed by an HMC, you must authenticate to the HMC. You can authenticate to the HMC by using the **hmcauth** command or, by defining an HMC object through network installation manager (NIM). The following characteristics apply to an HMC-based Live Update operation:

- The `hmcclientliveupdate` HMC role has all the privileges that are required for the Live Update operation. If a user is defined on the HMC with this role, the authentication can be done with this user rather than the `hscroot` user.
- When you run the Live Update operation, the value of the *lpar_id* attribute changes. You can request a specific value for the *lpar_id* attribute in the `lvupdate.data` file, but it cannot be the same as the original value.

PowerVC-based Live Update operation

If the LPAR is managed by PowerVC, you can authenticate with the PowerVC by using the **pvcauth** command or by defining a PowerVC object through NIM. The following characteristics apply to a PowerVC-based Live Update operation:

- When you run the Live Update operation, the value of the *lpar_id* attribute changes. But, you cannot request a specific value for the *lpar_id* attribute in the `lvupdate.data` file.
- If multiple profiles are associated with the LPAR, only the active profile is maintained by the Live Update operation. The other profiles are not preserved after the Live Update operation is complete.
- The virtual adapter ID values, also known as slot numbers, might change during the Live Update operation.

- The running workload must be able to accommodate the *blackout time*. The blackout time is the duration when the running processes are paused during the Live Update operation. The blackout time can be estimated by running the Live Update operation in the preview mode. Protocols such as transmission control protocol (TCP) use a back-off retransmit timeout that allows TCP connections to remain active during the blackout time, so the blackout time is not apparent to most workloads.
- The `bos.liveupdate` fileset must be installed to use the Live Update feature. This fileset is installed as part of the base AIX filesets, but could be missing if a migration installation was performed to migrate to AIX 7.2.
- The `dsm.core` and `dsm.dsh` filesets must be installed to use the Live Update feature with NIM.
- In the logical partition profile on the HMC, the minimum memory setting must be greater than or equal to 2 GB, which is the minimum amount of memory that is required to boot the AIX operating system.

Live Update restrictions:

Consider the following restrictions for the AIX Live Update operation:

I/O restrictions

- Any Coherent Accelerator Processor Interface (CAPI) device must not be open during the Live Update operation.
- No physical or virtual tape or optical device is supported. These devices must be removed before the Live Update operation can proceed.
- The **mirrorvg** utility can mirror up to 3 copies. If the root volume group of the original partition is already being mirrored with 3 copies, the Live Update operation cannot proceed.
- The Live Update operation is not supported on diskless AIX clients.
- The Live Update operation is not supported in a multibos environment.
- Data Management API (DMAPI) is not supported by the Live Update feature.

  Note: During the Live Update operation, IBM Spectrum Protect Space Management (HSM) cannot be in an active state because it uses DMAPI.
- The Live Update operation supports Virtual Small Computer System Interface (vSCSI) only for disks that are backed by physical volumes or by Shared Storage Pool (SSP) logical units. The vSCSI disks that are backed directly by logical volumes are not supported.
- If you run the **syncvg** command on non-rootvg volume groups during the Live Update operation, the operation might fail.

- When you create a new logical volume or extend a logical volume on rootvg during the Live Update operation, the **physicalvolume** parameter must be used. You must not use the **mhdisk** parameter that is specified in the `lvupdate.data` file. Otherwise, the Live Update operation might fail.
- If you run the **mount** command during the Live Update operation, the update might fail.
- After the Live Update operation is complete, if only interim fixes were applied, the `mhdisk` disk specified for the rootvg mirror volume group is labeled as `old_rootvg`. The `old_rootvg` volume group can be used for a reboot to return to the previous version of the root volume group before the update was applied.
- An existing `altinst_rootvg` label can cause the Live Update operation to fail.
- Geographic Logical Volume Manager (GLVM) is supported only within PowerHA® SystemMirror®.
- Network File System (NFS) mounts with Kerberos security are not supported.
- If Power Flash Caching is enabled (by using the **cache_mgt** command, for example), the caching is disabled during the Live Update operation, and re-enabled after the Live Update operation. The cached data is invalidated as a result, which can have a performance impact for some period of time until caching is resumed.
- If you create or delete file systems during the Live Update operation, the Live Update operation might fail.
- If you restart a Virtual I/O Server during a Live Update operation, the Live Update operation might fail.
- Adding or removing I/O adapters during the Live Update operation can cause the operation to fail.
- Increasing the size of a disk (for example, by using the `GROW LU` capability of Shared Storage Pools) during the Live Update operation can cause the operation to fail.
- An active Encrypted File System (EFS) mount point is not supported with the Live Update function.
- If you add or remove a paging space during the Live Update operation, the Live Update operation might fail.
- A Power Virtualization Center (PowerVC)-based Live Update operation might fail if a storage device is accessed by using a Fibre Channel over Ethernet (FCoE) adapter. PowerVC does not support FCoE network.

Security restrictions

- The Live Update operation is not supported when a process is using Kerberos authentication.
- The Live Update feature does not support PowerSC™ Trusted Logging.
- The Live Update feature is not supported if any of the following security profiles are active: high-level security (HLS), medium-level security (MLS), Sarbanes-Oxley (SOX) - Control Objectives for Information and Related Technology (COBIT), payment card industry (PCI) (any version), database, or Department of Defense (DoD) (any version).
- The Live Update feature is not supported when audit is enabled for a stopped workload partition (WPAR).
- The Live Update feature does not support Public-Key Cryptography Standards # 11 (PKCS11). The `security.pkcs11` fileset cannot be installed.
- The Live Update feature is not supported when the **Trusted Execution** option is turned on (**TE=ON**) and if any update must be applied. If only interim fixes are applied and the **Trusted Execution** option is turned on, the following **Trusted Execution** options in the **trustchk** command are not supported:
  - **TEP=ON**
  - **TLP=ON**
  - **CHKSHLIB=ON** and **STOP_UNTRUSTD=ON**
  - **TSD_FILES_LOCK=ON**
- The Live Update feature does not support Internet Protocol Security (IPSec). The Live Update operation fails if IPSec is started.

- The Live Update operation fails if the Virtual Trusted Platform Module (VTPM) is in use for PowerSC Trusted Boot.

Reliability, availability, and serviceability (RAS) restrictions

- It cannot perform system trace of the Live Update operation if channel 0 is already in use.
- The Live Update function is not supported when ProbeVue is running. The ProbeVue session must be stopped to run the Live Update operation.
- User storage keys are not supported in the Live Update environment.
- The system dump that is present on the root volume group of the original LPAR might not be available after a successful Live Update operation.
- If livedump operations are in progress, then a live update may fail.

Miscellaneous restrictions

- Live Update does not support rootvg on PowerPath.
- Any interim fix that you want to install must have the `LU CAPABLE` attribute, which means the interim fix must be compatible with the Live Update operation. The **emgr** command can display this attribute. Ideally, all the interim fixes can be applied with the Live Update operation, but there might be few exceptions.
- The destination of the interim fixes must be on the root volume group of the client partition in either `/`, `/usr`, `/home`, `/var`, `/opt`, or `/tmp` file systems.
- The volume group definitions must not be changed during a Live Update operation. The changes include the usage of the **chvg**, **extendvg**, **reducevg**, **mirrorvg**, **unmirrorvg**, **syncvg**, **varyonvg**, **varyoffvg**, **exportvg**, **importvg**, **reorgvg**, **redefinevg** commands.
- The NFS-mounted executables must not be running during a Live Update operation.
- Active WPARs must be stopped before the Live Update operation.
- RSCT Cluster Services are stopped during a Live Update operation, and then restarted before the Live Update operation completes.
- The Oracle Real Application Cluster (RAC) database and DB2® pureScale® database are not supported by the Live Update operation.
- A configuration with 16 MB page support is not allowed. The promoted (16 MB Multiple Page Segment Size (MPSS)) pages by Dynamic System Optimizer (DSO) are supported by the Live Update operation.
- The Live Update operation is supported when the DSO running, but DSO optimization is reset by the Live Update operation. The optimization begins again based on workload monitoring after the Live Update operation.
- The Live Update feature is not supported on a partition that participates in Active Memory™ Sharing (AMS).
- The Live Update feature is not supported on a partition with the remote restart capability enabled, but the Live Update feature is supported on a partition with the simplified version of the remote restart capability enabled.
- If a running process has been checkpointed at any time (legacy AIX checkpoint), the Live Update operation will fail.
- The Live Update feature is not supported when Advanced Accounting is active.
- The console must be closed before running the Live Update operation. The Live Update operation will fail if the console device is open for any process.
- A system firmware update during a Live Update operation can cause the update to fail.
- PowerVM® Partition Suspend feature is not supported during a Live Update operation.
- A process that has the `/dev/kmem` file or the `/dev/nvram` file open can cause the Live Update operation to fail.

- A process that has locked its text or data region (for example, by using the **plock()** subroutine) can cause the Live Update operation to fail.
- A process that has a file from the `/proc` file system open can cause the Live Update operation to fail.
- If any memory ranges are associated with the named resource sets on the system, those memory ranges are not preserved by the Live Update operation. Also, if any exclusive resource sets are defined on the system, the Live Update operation fails.
- When you plan a PowerVC-based Live Update operation on a partition that uses storage from an SSP that is multi-tiered, you must set the SSP default tier to the same tier from which the storage was allocated to the partition. Otherwise, the Live Update operation might fail. PowerVC can allocate storage only from the default tier.
- You must not start an HMC-based Live Update operation on a partition that is managed by PowerVC because an HMC-based Live Update operation causes issues when PowerVC manages partitions. If an HMC-based Live Update operation is started, you must stop managing the partition from PowerVC by using the **Unmanage** option, and then import the partition to be managed by PowerVC by using the **Manage Existing** option.
- If you do a PowerVC VM resize operation, or any other operation that can cause PowerVC to initiate a DR add or remove, during the time that Live Update is running on that VM, it may cause the Live Update operation to fail.
- The Live Update operation is not supported with products that use Shared Memory Transport Sockets (e.g. Xserver).
- Any change to the PowerVC configuration that restarts the HTTP service during a live update might cause the live update to fail.
- If you move the source or destination system in maintenance during the Live Update operation, the Live Update operation might fail.
- The Live Kernel Update (LKU) operation is not supported on a system on which the `ramdisk` device is mounted.
- Installing the `cas.agent` fileset that is used by the IBM® Systems Director causes the Live Update operation to fail.

Related information:

Hardware and software requirements for PowerVC Standard Edition

Best practices for the Live Update function:

Review these best practices before you start the AIX Live Update operation.

- When you run the Live Update operation, the current configuration of the Virtual I/O Server (VIOS) partitions are modified while the adapters are moved to the surrogate partition. Therefore, it is recommended to turn on the **Sync current configuration** option so that the current profile also gets updated. If the current profile is not being synchronized, use caution when you restart any VIOS partitions. If the configuration is modified, and you start a VIOS partition from a profile that does not match the current configuration, the AIX partitions might lose access to their adapters.
- Before you run the Live Update operation, save a copy of the current partition profiles on the Hardware Management Console (HMC), so that all the information is backed up in case it is needed in the future.
- If you plan to install updates by using the Live Update function, the updates are always committed. A copy of the system without the updates is not saved automatically. You must always take a viable backup of the system by using commands such as **alt_disk_copy** or **mksysb** before you apply updates so that you can return to the previous level if required.
- If you are using vSCSI disks and create a backup copy (`alt_rootvg`) of the root volume group (rootvg), the AIX Live Update operation might change the Logical Unit Addresses (LUA) of the disks. In this scenario, if you boot from the backup copy (`alt_rootvg`), the **lspath** command might display the disk paths that are missing. The disk paths that are missing were associated with the old LUA values. The missing disk paths do not cause any functional problems. You can run the **rmpath** command to remove

the disk paths and have the same number of disk paths that you had before you ran the Live Update operation. The following example displays the missing disk paths and running the **rmpath** command to remove the disk paths:

```
root@AIXmig / # lspath
Enabled hdisk5 vscsi1
Enabled hdisk3 vscsi1
Enabled hdisk4 vscsi1
Enabled hdisk0 vscsi1
Enabled hdisk1 vscsi1
Enabled hdisk2 vscsi1
Missing hdisk5 vscsi2
Enabled hdisk3 vscsi2
Missing hdisk4 vscsi2
Missing hdisk0 vscsi2
Enabled hdisk1 vscsi2
Missing hdisk2 vscsi2
Enabled hdisk6 vscsi1
Missing hdisk6 vscsi2
Enabled hdisk6 vscsi2
Enabled hdisk0 vscsi2
Enabled hdisk2 vscsi2
Enabled hdisk4 vscsi2

root@AIXmig / # rmpath -dl hdisk0 -p vscsi2
paths Deleted
root@AIXmig / # rmpath -dl hdisk2 -p vscsi2
paths Deleted
root@AIXmig / # rmpath -dl hdisk4 -p vscsi2
paths Deleted
root@AIXmig / # rmpath -dl hdisk5 -p vscsi2
paths Deleted
root@AIXmig / # rmpath -dl hdisk6 -p vscsi2
paths Deleted
```

- If you are using thin-provisioned Shared Storage Pool (SSP) storage, you must ensure that adequate real storage is available before you start a Live Update operation. The Live Update operation clones the root volume group that is used currently, and then creates a mirror copy of the root volume group that is used currently. If adequate real storage is not available, the Live Update operation fails.
- If you run the Live Update operation with a destination that is specified in the `lvupdate.data` file, then the partition might be temporarily moved to a different managed system. To preserve network connectivity during this time frame, all VLANs that are used on the partition must be supported by the physical network.
- If a Live Update operation fails, it automatically attempts to clean up the system. When the default cleanup fails, the following message prompts the user to run the **clvupdate** command:

  ```
  1430-115 Warning: Cleanup from the Live Update failure might not have been successful. Run clvupdate to finish the cleanup.
  ```

  If you are prompted with a warning message, run the **clvupdate** command to clean up the system before you attempt the Live Update operation again. Depending on the exact condition, the **clvupdate** command prompts the user with a set of actions to take to complete the cleanup.

Best practices for Live Update in PowerVC management

Review these best practices before you start the AIX Live Update operation on a partition that is managed by IBM Power Virtualization Center (PowerVC).

- If you plan to use the Live Update function on a partition that is managed by PowerVC, a backup copy of the system image without the interim fixes or updates is not saved automatically. You must always take a viable backup of the system image by using commands such as **alt_disk_copy** or **mksysb** before you apply updates so that you can return to the previous level of the system, if required.

- If an existing logical partition must be managed by PowerVC, verify that the boot volumes are set properly before you attempt a Live Update operation on the logical partition. When PowerVC imports the logical partition, PowerVC might not mark the correct volumes as boot volumes. Incorrect boot volumes can cause unexpected results when logical partitions are rebooted and can also cause the Live Update operation to fail.
- If you plan to use the Live Update function on a partition that is managed by PowerVC, you must set the **network_allocate_retries** property to a minimum value of 10 on the PowerVC. This property must be specified in the `/etc/nova/nova.conf` file on the PowerVC. If the property is not present in the file, you must add this property as a new line as shown in the following example:

  ```
  network_allocate_retries = 10
  ```

Configuring resources for Live Update:

You must configure the following resources for the AIX Live Update operation to complete successfully: CPU, memory, storage, I/O, and `lvupdate.data` file.

CPU and memory

The extra amount of CPU and memory resources that are required temporarily during the Live Update operation is equal to the amount of current resources that are used by the logical partition that must be updated with any interim fix installed. These CPU and memory resources must be available on the same frame when the Live Update operation is initiated, and are released by the time the Live Update operation completes. The following approaches reduce the impact of this requirement:

- Enable Capacity on Demand (CoD) resources during the AIX Live Update operation on hosts that are managed by the Hardware Management Console (HMC).

  If sufficient unlicensed and inactivated resources are available in the server that contains the logical partition that must be updated, the Live Update function automatically activates Enterprise Pool CoD resources until the Live Update operation is complete. Enterprise Pool CoD resources can be acquired in the following cases:

  – The compliance state of the pool must not be `out of compliance` according to your CoD licensing agreement.
  – If additional resources are activated, the total number of activated Enterprise Pool CoD resources must not exceed twice the number of entitled Enterprise Pool CoD resources.

  For other types of CoD resources, you must manually enable the CoD resources before you start the Live Update operation.

  Note: If you see `"1430-187 WARNING: Couldn't release the unreturned mobile CoD resources generated on CEC 'hostname'."`, check configuration of the Enterprise Pool, and make required updates to the resource allocation.
- Use Dynamic Logical Partitioning (DLPAR) to reduce the CPU and memory resources by half before the Live Update operation, and then increase them again when the Live Update operation is complete. This method impacts the performance of the partition during the Live Update operation, but it allows the operation to complete with no additional resources.
- Configure Live Update to allow the Live Update operation to run on another host. This process is only available if the host is managed by the IBM Power Virtualization Center (PowerVC). You must use the `lvupdate.data` file to indicate that the Live Update operation can be run on another host if the CPU or memory resources are insufficient on the initial host. To implement this process, the partition is automatically migrated by using the PowerVC Live Update operation. This migration process is temporary. You can use the *force_migration* option in the `lvupdate.data` file to force Live Update to run on another host, even if there is sufficient CPU and memory resources on the local host.

Storage

The Live Update operation requires at least 2 additional disks. The first disk (or set of disks) is required for the initial boot disk of the surrogate partition. This disk is shown as `lvup_rootvg` when you use the **lspv** command, and is not available for reuse until after the next Live Update operation, or after a system reboot. As part of the Live Update operation, an entry is added to the `/etc/inittab` file to remove the `lvup_rootvg` label on the disk (or set of disks) so the disk is available for general use after reboot. If the system is not rebooted, a subsequent Live Update operation will remove the label and the disk will be available for general use. The second disk (or set of disks) is required to create an additional mirror of the root volume group.

If the Live Update operation includes only interim fixes, this new mirror is not updated and is renamed to `old_rootvg` at the completion of the Live Update operation. In this case, this mirror copy can be used after the Live Update operation to move the system back to the previous level, if required, by rebooting the partition from this `old_rootvg` mirror. If any updates were applied with the Live Update operation, the new mirror includes the updates and is not named `old_rootvg`. In this case, it is a best practice to create a backup of the rootvg before you start the Live Update operation if you want to move the system back to the previous level.

If the partitions are managed by PowerVC, an `old_rootvg` mirror is not created by the Live Update operation. In this case, you can back up the rootvg before you start the Live Update operation if you want to move the system back to the previous level.

This disk can also be reused for another purpose. Depending on the system configuration, additional temporary disks might be required. If paging space is present on a non-rootvg disk, or if a memory dump device is present on the non-rootvg disks, two sets of disks must be provided (one set for the original partition and another set for the surrogate partition) with enough capacity for these paging spaces and memory dump devices. The preview mode of the Live Update operation can compute the required amount of space. These disks are available for reuse when the Live Update operation completes.

If the LPAR, which you want to update, is managed by an HMC, the required storage devices must be specified in the `disk` stanza of the `lvupdate.data` file. If the LPAR is managed by PowerVC, the PowerVC manages the storage devices and the disk names are not specified.

If the Live Update operation fails, it logs information in the `/var/adm/ras/liveupdate/logs` directory. This information might be required for service support. New log files are created in this directory with subsequent Live Update operations and the older log files are renamed to include the timestamp in its names. These older log files can be removed, if required, to free some space.

Reliability, availability, and serviceability (RAS) information that is associated to the Live Update operation is available in the `/var/adm/ras/liveupdate` directory. Component traces are available in the `ct_dump` directory, and the lightweight memory traces are available in the `lmt_dump` directory. If the Live Update trace is enabled, the`trcfile_orig` file contains traces for the original node and the `trcfile_surr` file contains traces for the surrogate node. Live dumps during the Live Update operation are collected in the `/var/adm/ras/livedump` directory.

If any service issue occurs with the Live Update operation, the **snap -U** command collects all the required information for the support team.

I/O

All I/O must be virtualized through Virtual I/O Servers (VIOS) for the Live Update operation. All VIOS slot numbers are the same on both the VIOS servers and the client when the Live Update operation completes. At least two paths must exist to all disks. Half of the paths are removed from the original partition and used from the surrogate partition during the Live Update operation, and all paths are moved to the surrogate partition before the Live Update operation completes. The Live Update operation can work with the following multipathing solutions: IBM AIX Multipath I/O and IBM Subsystem Device Driver Path Control Module (SDDPCM).

There are some device Object Data Manager (ODM) attributes that can be changed but the new values do not take effect until the next system reboot. Since the Live Update operation acts as a system reboot, any such attributes take effect as a result of the Live Update operation.

lvupdate.data file

When you perform a Live Update operation, the **geninstall** command searches for a stanza file that is called `lvupdate.data` in the `/var/adm/ras/liveupdate` path. This file contains the appropriate input data for the Live Update operation. The `/var/adm/ras/liveupdate/lvupdate.template` file contains the latest descriptions of all possible fields. The following example is a sample `lvupdate.template` file that contains description of the basic fields:

```
| #
| # The lvupdate.template file can be used to create the
| # /var/adm/ras/liveupdate/lvupdate.data file, which is
| # required for Live Update (geninstall -k ... ).
| # If the LPAR that you want to update is managed by HMC, the pvc stanza does not
| # apply and it must not be specified.
| # If the LPAR that you want to update is managed by PowerVC, the disk and hmc stanzas
| # do not apply and these stanzas must not be specified.
| # All fields in the disk stanza can be one disk or a comma-separated
| # list of disks.
| #
| # If preview is entered as part of the geninstall command_line or
| # in the SMIT menus, then no lvupdate.data file is required. If one is
| # provided, and the disk stanza completed, then size checking on the
| # disks will be performed.
| #
| # general:
| #      kext_check =  <yes | no>  Blank defaults to yes.  If no, the Live Update
| #         operation will be attempted regardless as to whether all the loaded
| #         kernel extensions are determined to be safe or not.
| #      cpu_reduction = <yes | no> Blank or omitted defaults to no. If yes, the
| #         live update operation will, in the absence of sufficient processing
| #         capacity to create and boot the surrogate normally, determine whether
| #         removing some capacity from the original would result in sufficient
| #         available capacity, and if so, will automatically remove
| #         the capacity from the original (later to restore the surrogate to full
| #         capacity once the original is deleted).
| #
| # disk:
| #      nhdisk = <disk1,disk2,...>  The names of disks to be used to make a copy
| #         of the original rootvg which will be used to boot the Surrogate
| #         (surr-boot-rootvg). The capacity needs to match the capacity of the
| #         "required" file systems (/, /var, /opt, /usr, /etc) from the
| #         orig-rootvg. (If previewing, size checking will be performed.)
| #      alt_nhdisk = <disk1,disk2,...>   The names of disks to be used if the disks
| #         specified for the nhdisk attribute are not currently available
| #         to be used by Live Update. The capacity requirements are
| #         the same as nhdisk.
| #      mhdisk = <disk1,disk2,...>  The names of disks to be used for the
| #         mirrored rootvg (surr-mir-rootvg) on the Surrogate. The capacity needs
| #         to match the capacity of orig-rootvg. After the live update, the
| #         surr-mir-rootvg remains as a copy of the rootvg from before the
| #         updates were applied. (If previewing, size checking will be
| #         performed.)
| #      tohdisk = <disk1,disk2,...>  The names of disks to be used as temporary
| #         storage for the Original. This is only required if the Original
| #         is using paging space or dump devices on non-rootvg volume groups. The
| #         capacity needs to match the total capacity of paging spaces and dump
| #         devices defined on non-rootvg volume groups for the original
| #         partition. (If previewing, size checking will be performed.)
| #      tshdisk = <disk1,disk2,...>  The names of disks to be used as temporary
| #         storage for the Surrogate. This is only required if the Original is
| #         using paging space or dump devices on non-rootvg volume groups. It
| #         must have the same capacity as tohdisk. (If previewing, size checking
```

```
#         will be performed.)
#
# hmc:
#      lpar_id = <lpar id>  Indicates the desired partition id for the
#         Surrogate.
#      alt_lpar_id = <lpar id>  Indicates an alternate partition ID for the
#         Surrogate. If the value specified for the 'lpar_id' attribute is already in use,
#         Live Update will use this alternate ID if it is not in use.
#      management_console = <HMC IP Address>
#      user = <HMC user>  Indicates the user ID that is used to access HMC.
#      storage_template_override = <storage template name>  Indicates the name
#
# pvc:
#      management_console = <hostname or IP Address of the server hosting the PowerVC identity service>
#      user = <PowerVC user>   Indicates the user ID that is used to access PowerVC.
#      project = <PowerVC project>   Indicates the project name that is used to access PowerVC.
#      If this attribute is not specified, the Live Update operation uses the ibm-default project.
#      storage_template_override = <storage template name>  Indicates the name
#      of the storage template that must be used for the boot volume of the surrogate partition.
#      This parameter is optional.
#      If this parameter is specified, the Live Update operation uses the specified storage template
#      for the boot volume of the surrogate partition. If this parameter is not specified,
#      the Live Update operation uses the storage template of the original root
#      volume group, if any. If a storage template is not associated with the original root volume group,
#      the default storage template of the rootvg storage provider is used for the boot volume
#      of the surrogate partition.
#      destination = < PowerVC host name | ANY >  This attribute is used to
#         specify the host on which the Live Update operation will be executed
#         if the resources are insufficient on the local host, or if the
#         force_migration attribute is specified with a value equals to yes.
#         If the attribute is set to ANY, the destination will be selected
#         according to the placement policy of the host group in PowerVC.
#         If this parameter or its value is not specified, the Live Update
#         operation is executed locally.
#      force_migration = < yes | no >  When set to yes, the Live Update
#         operation is executed on the host specified by the destination
#         attribute even if the resources are sufficient locally.
#         This parameter is optional and can be set to yes only when a
#         destination is specified.
#
# trace:
#      trc_option = <trace command options>  This can be a hook id
#         with -j hookid1,... or any other trace option.
#         If specified, the Live Update commands will be traced using
#         the specified options.  One or more can be specified.
#         If the stanza is present in the lvupdate.data file,
#         with a blank trc_option field, the default parameters
#         "-a -U -C and -o" are used to trace the Live Update commands.
#         Users need not provide redundant options such as "-a -U -C and -o"
#         in the trc_option field for trace stanza.
#         Do not add a trace stanza to the lvupdate.data file unless you
#         want the Live Update commands to be traced.
#

general:
         kext_check =

disks:
         nhdisk  =
         mhdisk  =
         tohdisk =
         tshdisk =

hmc:
         lpar_id  =
         management_console =
         user =
```

Related information:
Power Enterprise Pool compliance

Prerequisites for Live Update:

The following minimum levels of these system components are required for the AIX Live Update function:

Prerequisites for Live Update when LPAR is managed by HMC:

System firmware

- Ax730_066 (Limitation: It does not allow PowerVC to seamlessly manage the updated LPAR)
- Ax740_043 (Limitation: It does not allow PowerVC to seamlessly manage the updated LPAR)
- Ax770_063
- Ax773_056
- Ax780_056

Hardware Management Console (HMC)
840

Note: Either HMC or PowerVC is required for the AIX Live Update function.

Virtual I/O Server
2.2.3.50

RSCT (if required)
3.2.1.0

PowerHA (if required)
7.2.0

PowerSC (if required)
1.1.4.0

Subsystem Device Driver Path Control Module (SDDPCM) (if required)
2.6.7.0

Prerequisites for Live Update when LPAR is managed by PowerVC:

System firmware

- Ax770_063
- Ax773_056
- Ax780_056
- Ax840
- Ax860

Hardware Management Console (HMC - Either HMC or NovaLink is required when PowerVC is used)

- 860 SP2
- 870
- 910

IBM Power Virtualization Center (PowerVC)
1.3.3.1

NovaLink (Either NovaLink or HMC is required when PowerVC is used)
1.0.0.6

Virtual I/O Server
2.2.6.0

RSCT (if required)
3.2.3.0

PowerHA (if required)
7.2.0

PowerSC (if required)
1.1.4.0

Subsystem Device Driver Path Control Module (SDDPCM) (if required)
2.6.7.0

Preview mode:

To validate the system configuration for the AIX Live Update operations, you can use the preview mode before you attempt the Live Update operation. Running the preview mode ensures that both the environment and the parameters that are specified in the `lvupdate.data` file meet the requirements for the Live Update operation. The preview mode report also provides estimates of the time that is required for the complete Live Update operation, along with the amount of time for which the processes will be paused (the blackout period) based on the workload that is running during the preview mode.

If the `lvupdate.data` file does not exist, or if the required disks are not specified, the preview mode reports the storage that is required for Live Update operation. If the `lvupdate.data` file specifies the required disks, the preview mode validates the sizes.

Performing the Live Update operation by using NIM

Network Installation Manager (NIM) can be used to start an AIX Live Update operation on a target machine either from a NIM master (also known as central master) or from the NIM client. The required authentication to the HMC can also be managed within the NIM framework by defining a Hardware Management Console (HMC) object. Similarly, for a NIM client that is managed by IBM Power Virtualization Center (PowerVC), the authentication can be managed by defining a PowerVC object in NIM.

Use the following step-by-step examples to set up NIM and to set up the updates that are initiated from either the client or the master:

1. Generate the HMC password key.
   ```
   # /usr/bin/dpasswd -f /export/eznim/passwd/hmc_passwd -U hscroot -P abc123
   ```
2. Use this key to define an HMC object.
   ```
   # nim -o define -t hmc -a if1="find_net hmc_object 0" -a net_definition="ent 255.255.255.0 9.1.2.1"
                   -a passwd_file=/export/eznim/passwd/hmc_passwd hmc_object
   ```
3. Define managed system of NIM stand-alone machine.
   ```
   # nim -o define -t cec -a hw_type=8203 -a hw_model=E4A -a hw_serial=0123456
                   -a mgmt._source=hmc_object cec1
   ```
4. Exchange Secure Shell (SSH) keys between HMC and NIM master.
   ```
   # dkeyexch -f /export/eznim/passwd/hmc_passwd -I hmc -H hmc_object
   ```
5. Define the NIM stand-alone machine that points to the Central Electronic Complex (CEC).
   ```
   # nim -o define -t standalone -a if1=find_net mac1 0" -a net_definition="ent 255.255.255.0 9.1.2.1"
                   -a net_setting1="100 full" -a mgmt_source=cec1 -a identity=<lpar_id> client1
   ```

Note: The Live Update operation started by NIM calls the **hmcauth** command during the **cust** operation to authenticate to the NIM client with the HMC by using the HMC `passwd` file.

Starting the Live Update operation from NIM master

To use a NIM `live_update_data` resource, run the following command:

```
# nim -o cust -a live_update=yes -a live_update_data=lvup -a lpp_source=720lpp
               -a filesets=IZ12345.140806.epkg.Z client1
```

To use the client's `/var/adm/ras/liveupdate/lvupdate.data` file, run the following command:

```
# nim -o cust -a live_update=yes -a filesets=IZ12345.140806.epkg.Z client1
```

To run the Live Update operation in preview mode, run the following command:

```
# nim -o cust -a live_update=yes -a live_update_data=lvup -a install_flags="-p"
               -a lpp_source=720lpp -a filesets=IZ12345.140806.epkg.Z client1
```

Starting the Live Update operation from NIM client

To use separate operations to allocate and run the Live Update operation, run the following command:

```
# nimclient -o allocate -a lpp_source=720lpp -a live_update_data=lvup
# nimclient -o cust -a live_update=yes -a filesets=IZ12345.140806.epkg.Z
```

To allocate and run the Live Update operation together, run the following command:

```
# nimclient -o cust -a live_update=yes -a lpp_source=720lpp -a live_update_data=lvup
                     -a filesets=IZ12345.140806.epkg.Z
```

To run the Live Update operation in preview mode, run the following command:

```
# nimclient -o cust -a live_update=yes -a lpp_source=720lpp -a live_update_data=lvup
                     -a install_flags="-p" -a filesets=IZ12345.140806.epkg.Z
```

Related concepts:

"Defining a live_update_data resource" on page 245
You can use the following command-line syntax and attributes to define a `live_update_data` resource.

Related information:

nimclient Command

nim Command

Performing the Live Update operation by using the geninstall command

After the `/var/adm/ras/liveupdate/lvupdate.data` file is created, you can use the **geininstall** command to initiate a AIX Live Update operation.

Use the following steps to start the Live Update operation by using the **geninstall** command:

1. If the logical partition (LPAR) is managed by an HMC, authenticate HMC.

   ```
   # hmcauth -u hscroot -a hmc_name
   ```

2. If the LPAR is managed by PowerVC, authenticate PowerVC.

   ```
   # pvcauth -u root -a powervc_host
   ```

3. Run the Live Update operation in preview mode.

   ```
   # geninstall -k -p -d /tmp IZ12345.140806.epkg.Z
   ```

4. Run the Live Update operation for the specified type of update.
   - To install an interim fix, run the following command:

     ```
     # geninstall -k -d /tmp IZ12345.140806.epkg.Z
     ```

   - To install the updates to two filesets and to install an interim fix, run the following command:

     ```
     # geninstall -k -d /tmp bos.mp64 bos.rte.libc IZ12345.140806.epkg.Z
     ```

   - To install all the updates and interim fixes that are available in the `/tmp/source` directory, run the following command:

     ```
     # geninstall -k -d /tmp/source all
     ```

- To install all the updates, but not interim fixes, that are located in the `/tmp/updates` directory, run the following command:

  ```
  geninstall -k -d /tmp/updates update_all
  ```

Note: You can install any updates and interim fixes by using your preferred methods, and then perform a Live Update operation instead of restarting the system by running the following command:

```
# geninstall -k
```

Related information:

geninstall Command

hmcauth Command

Advanced customization for Live Update

For some applications or kernel extensions, additional steps are required for seamless support of the AIX Live Update function. The following information is provided for independent software vendors (ISVs) or custom application developers who need to use the utilities and frameworks that are provided with the Live Update feature.

Notification frameworks:

Most applications do not need to be aware of the AIX Live Update operation. During the Live Update operation, an application is checkpointed after the application receives a checkpoint signal. During the checkpointing process, the mobility mechanism takes over the application and saves the application-specific resources, then re-creates the application on the surrogate partition. When the resources are restored, the application resumes its operations. All applications are checkpointed at the same time and restarted at the same time.

Some applications need to interact with the Live Update operation. Such applications can use the Dynamic Logical Partitioning (DLPAR) framework. When the Live Update operation starts on the original partition, applications are notified during the *check* phase. The applications can use the `dr_reconfig()` system call to acknowledge the Live Update operation before the Live Update timeout (60 seconds). This timeout provides time to applications to prepare itself for the DLPAR event.

During the *check* phase, an application can query the `dr_info` structure for details about the DLPAR event such as the type of event and the current phase. For the Live Update event, the origin of the notification (the original partition or the surrogate partition) can also be queried. An application can use a DR_EVENT_FAIL event to stop the Live Update operation during the *check* phase, if the application cannot survive a checkpoint or restart at that time. Due to the timing of the *check* notification on the surrogate partition, the DR_EVENT_FAIL event applies to only those applications that are started from the `inittab` process on the `surr-boot-rootvg` volume group.

Before the applications are checkpointed on the original partition, a DLPAR notification is sent to applications during the *pre* phase. When the mobility operation is done and applications are restarted on the surrogate partition, a DLPAR notification is sent to applications during the *post* phase at both the original and surrogate partitions. Only base processes can see the *post* event on the original partition. Applications that are moved to the surrogate partition receive the *post* notification in the surrogate partition. If an error occurs, a DLPAR notification is sent to the applications during the *post-error* phase.

Dynamic reconfiguration or DLPAR framework

The Live Update operation is registered as a Dynamic Reconfiguration (DR) or Dynamic Logical Partitioning (DLPAR) operation. It means that when the Live Update operation is running, no other DLPAR operation can be performed, and when any DLPAR operation is in progress, the Live Update operation cannot be started. Therefore, the configuration of the original LPAR is preserved during the Live Update operation. The DLPAR operations resume after the Live Update operation completes.

The DLPAR framework is also used to inform applications, kernel, and kernel extensions of the Live Update operation. The DLPAR framework supports the following phases:

- check
- pre
- post
- post-error

A notification is sent to applications, kernel, or kernel extensions at each of these four phases. If applications and kernel extensions are integrated into the DLPAR framework, the applications and kernel extensions can interact with the Live Update operation.

Integration with DLPAR

The applications integrate with the DLPAR framework in the following methods: By handling the SIGRECONFIG signal. Within the signal handler, the **dr_reconfig()** subroutine can be used to query and acknowledge the DLPAR event. The handler must reconfigure the application.

Another method is to install a set of DLPAR scripts. These scripts are started when a DLPAR event occurs, and must be designed to respond to Live Update operation aptly. Applications must reconfigure itself when they receive DLPAR notification.

Kernel extensions use the **reconfig_register_list()** kernel service to register reconfiguration handlers for DLPAR events. These handlers are called when DLPAR events occur.

Live Update support in DLPAR

The Live Update operation introduces a new DLPAR event.

The `dr_op` field of the `dr_info` structure is set to DR_OP_LVUPD for a Live Update event. The field in the `dr_info` structure that indicates the origin of the DLPAR notification is defined in the `sys/dr.h` file as follows:

```
ushort lvup
```

When the **dr_reconfig()** subroutine is called for the Live Update event, the `lvup` bit is set to LIVEUPDTORIG (the original partition is the origin of the DLPAR notification) or LIVEUPDTSURR (the surrogate partition is the origin of the DLPAR notification). These values are defined in the `dr.h` file as follows:

```
#define LIVEUPDTORIG   0x1
#define LIVEUPDTSURR   0x2
```

Alternative to DLPAR

The DLPAR or DR framework does not enforce an order of execution of scripts within the same phase. If the subsystems rely on synchronization of their operations during a specific phase, these subsystems must implement the synchronization among itself.

To save these subsystems from having to implement a synchronization mechanism, the Live Update framework provides an alternative notification system. The **lvupdateRegScript** command can be used to register a specific script with a priority.

The priority can be an integer value in the range 1 - 10. For more information about priorities, see the timeline table in the “Timeline to run the DLPAR scripts” on page 393 topic. During the Live Update operation, before the *check* event is issued, the scripts that are registered with the LVUP_CHECK event

are executed; the order of execution starts with scripts with the highest priority to the lowest priority. The same methodology is applied to the rest of the phases. The script must be registered only once, during the installation of the application.

The script owner must specify whether the script must be registered and run on the original partition or the surrogate partition. The Live Update operation fails if a script fails during the LVUP_CHECK or LVUP_PRE events.

Related information:

lvupdateRegScript Command

dr_reconfig System Call

Actions taken by DLPAR scripts

reconfig_register_list() and reconfig_complete() Kernel Service

Timeline to run the DLPAR scripts:

The AIX Live Update notifications are run on both original and surrogate partitions.

The order of execution of the phases is as follows:

Original node	**Surrogate node**
LVUP_CHECK Priority 1 If error occurs, LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation** This script is invoked regardless of phase run.	
...	
LVUP_CHECK Priority 10 If error occurs, LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
DR_CHECK; Return code (RC) is checked. If error occurs, DR_POST_ERROR LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
Alternate disk install	**Boot the surrogate node**
	DR_CHECK; RC is checked. If error occurs, DR_POST_ERROR **Terminate Live Update operation**
	LVUP_CHECK Priority 10 If error occurs, DR_POST_ERROR LVUP_ERROR Priority 1 ... LVUP_ERROR Priority 10 **Terminate Live Update operation**

Original node	Surrogate node
	...
	LVUP_CHECK Priority 1 If error occurs, DR_POST_ERROR LVUP_ERROR Priority 1 ... LVUP_ERROR Priority 10 **Terminate Live Update operation**
If error occurs on the surrogate node, DR_POST_ERROR LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
LVUP_PRE (applications) Priority 1 If error occurs, LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
...	
LVUP_PRE (applications) Priority 10 If error occurs, LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
DR_PRE (applications); RC is not checked. This DR_PRE script is run for the processes that are being migrated.	
Mirror volume group	
	DR_PRE (applications); RC is not checked. This DR_PRE script is run for the running base processes.
	LVUP_PRE (application) Priority 10 This DR_PRE script is not run for the base process, but it is available for the migrated processes. If error occurs, LVUP_ERROR Priority 1 ... LVUP_ERROR Priority 10 **Terminate Live Update operation**
	...
	LVUP_PRE (application) Priority 1 If error occurs, LVUP_ERROR Priority 1 ... LVUP_ERROR Priority 10 **Terminate Live Update operation**
If error occurs on the surrogate node, DR_POST_ERROR LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	

Original node	Surrogate node
Applications are frozen, network is blocked	
LVUP_PRE (Kernel) Priority 1 If error occurs, LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
...	
LVUP_PRE (Kernel) Priority 10 If error occurs, LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
DR_PRE (Kernel); RC is checked. If error occurs, DR_POST_ERROR LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation** The DR_POST_ERROR script is run once for both the kernel and application DR_PRE scripts.	
	If error occurs on the original node, DR_POST_ERROR LVUP_ERROR Priority 1 ... LVUP_ERROR Priority 10 **Terminate Live Update operation** The DR_POST_ERROR script is run once for both the kernel and application DR_PRE scripts.
Split volume group	
	Import volume group
	DR_PRE (Kernel); RC is checked. If error occurs, DR_POST_ERROR
	LVUP_PRE (Kernel) Priority 10 If error occurs, DR_POST_ERROR LVUP_ERROR Priority 1 ... LVUP_ERROR Priority 10 **Terminate Live Update operation**
	...
	LVUP_PRE (Kernel) Priority 1 If error occurs, DR_POST_ERROR LVUP_ERROR Priority 1 ... LVUP_ERROR Priority 10 **Terminate Live Update operation**

Original node	Surrogate node
If error occurs on the surrogate node, DR_POST_ERROR LVUP_ERROR Priority 10 ... LVUP_ERROR Priority 1 **Terminate Live Update operation**	
	Migrated applications are unfrozen, network is unblocked
	Point of no return
DR_POST script is run; RC is not checked. (For cleanup of files if the Live Update migration is not acceptable to the customer.)	DR_POST script is run; RC is not checked. (For migrated applications.)
LVUP_POST (application) Priority 1	LVUP_POST (application) Priority 10
...	...
LVUP_POST (application) Priority 10	LVUP_POST (application) Priority 1
	Original LPAR deleted, Surrogate UUID updated
	LVUP_COMPLETE (application) Priority 10
	...
	LVUP_COMPLETE (application) Priority 1

System tunable parameters:

The AIX Live Update operation must ensure that the tunable parameters are set on the surrogate partition similarly as they were set on the original partition. When the Live Update operation is started, it captures the currently set tunable parameters and their parameters. Therefore, these tunable parameters must not be changed during the Live Update operation if these tunable parameters are to be preserved in the surrogate partition. The tunable parameters configuration is set on the surrogate partition when the surrogate partition is started.

The AIX Runtime Expert (artex) mechanism is used to capture and set tunable parameters.

Related information:

AIX Runtime Expert

Application customization for Live Update:

The AIX Live Update operation performs the classification only during the Live Update operation.

Base processes

A *base process* is a process that does not participate in the Live Update operation. This process is not frozen or checkpointed. It does not have access to the network or data storage during the mobility phase of the Live Update operation. Base processes can be grouped as follows:

- A set of core AIX services that continues to run in the Live Update operation after applications are checkpointed on the original partition and before they are restarted on the surrogate partition. These services are necessary because the memory associated with the moved processes is transferred asynchronously after they are checkpointed. So the original partition must be functional until all the memory is moved. All processes that are attached to the console of the original partition are marked as base processes because the console must remain associated with the original partition.
- A set of services that are required to boot the surrogate partition to the point that it can communicate with the original partition and receive the moved processes. A customized `/etc/inittab` file is used to determine the services that start on the surrogate partition.

Mobile processes

Processes that are moved from the original partition to the surrogate partition as a part of the Live Update operation. All processes apart from the base processes are called *mobile processes*. Most workload processes are mobile processes. A mobile process has the same process ID (pid) or thread ID (tid) at the end of the Live Update operation. The mobile processes can be classified into the following groups:

- **Checkpointable processes**: These processes are frozen and their state is checkpointed on the original partition. These processes are re-created on the surrogate partition.
- **Exit processes**: These processes are frozen on the original partition. The Live Update operation does not checkpoint the state of these processes. These processes are re-created on the surrogate partition, but instead of restarting them at the instruction where they were checkpointed, they are forced to call the **exit()** function and terminate. Applications that are not impacted by its state when restarted can choose this method. These applications do not have to release resources that are not supported by the mobility operation. When these applications are monitored by a daemon mechanism (such as `init` or `srcmstr`), a new instance is started in the surrogate partition after they exit from the original partition.

For the Live Update operation to succeed, the following rules must be followed by the processes in the system:

- A `kproc` is a base process.
- The `init` process is a base process.
- A direct child of `init` can be either a base process or a mobile process.
- Children of a base process other than the `init` process are base processes.
- A base process is either a direct child of the `init` process or the child of another base process.
- Base processes do not share resources with non-base processes.

An application can register its processes as base process or exit process by using one of the following methods:

Static registration

Processes are registered by using the **lvupdateSetProcs** command. During the validation phase, the Live Update operation ensures that the rules for base processes are applied.

Dynamic registration

A process can register itself as a base process by using the **proc_mobility_base_set()** system call, or as an exit process by using the **proc_mobility_restartexit_set()** system call. The dynamic registration can occur only after the *check* notification is sent to the process. The system call ensures that the caller satisfies the base processes rules. Any existing child processes are automatically marked as a base process.

Inittab processes and init

When the `surr_boot_rootvg` volume group is cloned from the `orig_rootvg` volume group, the `/etc/inittab` file is replaced with a minimal set, which is designed for the Live Update operation. The following example shows a sample `initab` file:

```
:inittab.sur - live os update
init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
tunables:23456789:wait:/usr/sbin/tunrestore -R > /dev/console 2>&1 # Set tunables
securityboot:2:bootwait:/etc/rc.security.boot > /dev/console 2>&1
opt:2:wait:/usr/sbin/mount /opt
random:2:wait:/usr/sbin/randomctl -l
cons:0123456789:respawn:/usr/sbin/getty /dev/console
syslogd:2:once:/usr/sbin/syslogd >/dev/console 2>&1
slvupdate:2:wait:/usr/sbin/slvupdate >/dev/console 2>&1
ha_star:h2:once:/etc/rc.ha_star >/dev/console 2>&1
```

When the surrogate partition boots, it runs with the minimal set of processes to allow the Live Update operation to proceed.

Applications that prefer to be started as a part of the surrogate partition, can register to be included in the `/etc/inittab` file that is deployed on the `surr-boot-rootvg` environment by using the **lvupdateInit** command. These applications started from the `inittab` process are marked as base processes, hence these applications are not eligible to participate in any subsequent Live Update operation. In this case, the related kernel extensions must be loaded by the application started from the `inittab` process. These applications start in a running state on the `surr-boot-rootvg` environment. The **chroot** operation (changing the root directory) is performed on applications that are migrated from the original partition to the `/old` directory to run on the `surr-mir-rootvg` environment. The base applications that are started from the `inittab` process can access or chroot to the `surr-mir-rootvg` environment after the *post* notification.

Related information:

lvupdateSetProcs Command

lvupdateInit Command

proc_mobility_base_set Subroutine

proc_mobility_restartexit_set Subroutine

Kernel extension customization:

During the AIX Live Update operation, kernel extensions can be impacted. The Dynamic Logical Partitioning (DLPAR) platform is used to communicate the operation progress between the Live Update operation and kernel extensions.

The following table describes the kernel extension states in the original partition and the surrogate partition during each phase:

Phases	Original partition	Surrogate partition
check	Kernel extensions are notified at the same time as applications. Any data on the `orig-rootvg` environment is copied to the `surr-boot-rootvg` environment when the data is created.	Kernel extensions are notified at the same time as applications. Checkpointed data is available on both the `surr-boot-rootvg` and `surr-mir-rootvg` volume groups because of mirroring. The `surr-mir-rootvg` device is available only after the *pre* phase.
pre	Kernel extensions are notified after applications are checkpointed. The checkpointed data must be saved to the `orig-rootvg` volume group. Because of mirroring, the data is also available on the `surr-mir-rootvg` volume group. The data becomes available in the chrooted environment for the surrogate partition after the **splitvg** operation that occurs only after DLPAR notification. After a restart of the surrogate partition, kernel extensions need to account for the change of the location of the file. If the old path is `x`, the new path is `/old/x`.	Kernel extensions are notified when file systems of the `surr-mir-rootvg` volume group are mounted. The data that is collected on the original partition's *pre* phase is available only in the chrooted environment (after the root directory is changed). Applications that are on surrogate partition must be aware of the availability of the chrooted environment.
post	This notification is sent to applications when applications are started on the surrogate partition.	This notification is sent to applications when applications are started on the surrogate partition.
post-error	Kernel extensions can take appropriate action.	Gives kernel extensions the opportunity to respond to the Live Update failure depending on in which phase the *post-error* occurs.

If a kernel extension expects that the DLPAR handling operation takes a long time, the handler must return DR_WAIT to the caller, and proceed with the request asynchronously. When the request is completed, the handler must call the **reconfig_complete()** kernel service.

Application state located in kernel extensions must be considered from the related kernel extensions. The related kernel extensions need to checkpoint such application states when the applications are checkpointed and reload them with the right state when the applications are restarted.

Device considerations

When the surrogate partition is started, the devices must be configured similar to the configuration on the original partition. The same device on the original partition and the surrogate partition must have the same name, the same device number (`devno` (major, minor)), and the same device configuration.

Some devices might have customized attributes that are modified in Object Data Manager (ODM), but not taken effect (these changes take effect at reboot time of the LPAR). When the surrogate partition is booted, the customized attributes take effect. The storage devices might not have the same multipathing topology on the surrogate partition as the original partition.

Kernel extensions in mobility

Kernel extensions need special considerations for mobility so that the workload is not interrupted. For most kernel extensions, unloading them on the original partition and reloading them on the surrogate partition suffice.

Safe kernel extensions

By default, all kernel extensions that are loaded on the original partition must be identified as *safe* for the Live Update operations unless you have overridden it with the `kext_check` setting in the `/var/adm/ras/liveupdate/lvupdate.data` file.

Generally, a kernel extension is *safe* for the Live Update operation if the kernel extension is aware of the Live Update operation or does not need to be aware of the Live Update operation. A kernel extension is deemed to be Live Update *safe* if it meets one of the following requirements:

- The kernel extension is loaded with the **SYS_LUSAFE** flag.
- The kernel extension name is in the `/etc/liveupdate/lvup_SafeKE` file.

To mark the kernel extension as Live Update safe, the kernel extensions can be loaded by using the **sysconfig()** call with the **SYS_LUSAFE** flag that is defined in the `sys/sysconfig.h` file.

In some safe kernel extensions, the **SYS_LUSAFE** flag might not be set. You can mark them as safe for a Live Update operation by using the **lvupdateSafeKE** command.

Safe kernel extensions are listed in the `/etc/liveupdate/lvup_safeKE` file. Duplication is not allowed in this list. Each kernel extension must be listed with its full path.

In all modes, it is always validated that the loaded kernel extensions are safe, even when you choose not to enforce the requirement. In this case, the Live Update operation logs the non-compliant kernel extensions, but continues to operate.

Loading kernel extensions

When the surrogate partition is started, it loads only those kernel extensions that are related to devices that are configured. Normal commands that usually start during the regular initialization of an LPAR might not start. As a result, some kernel extensions that are needed by checkpointed applications might not be loaded when the applications are restarted. The Live Update framework offers more than one mechanism to handle such situation:

- Applications with kernel extensions can be enabled for checkpoint if they manage the loading and unloading of the kernel extensions. The unloading must occur before the freezing of the applications and you can load the kernel extensions when applications are restarted.
- Kernel extensions can be preloaded on the surrogate partition before the applications are restarted. The Live Update framework offers a registration mechanism. All loading methods that are registered for

the Live Update operation are executed before the applications are restarted. The **lvupdateRegKE** command can be used to add or remove kernel extensions to be preloaded.
- The full path of the kernel extension is needed. In a loading error, the Live Update operation is stopped.

Example for interaction between a process and a kernel extension

This example shows how the interaction between a process and a kernel extension must be handled. The goal of the Live Update operation is to preserve the behavior of workloads in the update process.

Suppose that an application comprises a `test_process` process and a `test_ke` kernel extension. The `test_ke` kernel extension has a variable counter that is used to count some events. The `test_process` process reads counter from `test_ke` and consumes it during its execution. When `test_ke` is loaded, the counter is initialized to 0. The counter's value increases with time. In the Live Update operation, when `test_process` is checkpointed, its process state is saved, but the counter value is not saved. Since kernel extensions are not checkpointed, you must ensure that the counter is preserved when it is loaded on the surrogate partition. This function is supported by the DLPAR framework in the Live Update operation.

1. Applications are checkpointed on the original partition.
2. A notification is sent to the kernel extensions at the *pre* phase.
3. The `test_ke` kernel extension uses the **reconfig_register_list()** kernel service to register reconfiguration handlers for DLPAR events.
4. In the handler for the *pre* phase, the counter is saved in the `/var/adm/ras/liveupdate/kext/test_ke` file. This file is located on rootvg so that it can be transferred to the surrogate partition after the partition is mirrored.
5. On the surrogate partition, the *pre* phase is sent to kernel extensions after the `surr-mirr-rootvg` environment is mounted. It means that the saved data for the `test_ke` kernel extension including variable counter is now available. The state of the `test_ke` kernel extension can be reconfigured to match the state when it was saved.

Related information:

lvupdateSafeKE Command

lvupdateRegKE Command

reconfig_complete() and reconfig_register_list() Kernel Services

Software product packaging

The following is additional software product packaging information.

Installing variously formatted software packages

You can use this information to install software packages received in different formats.

You can install RPM Package Manager (RPM), interim fix, and **InstallShield MultiPlatform** (**ISMP**) formatted packages in addition to **installp** formatted packages. Use the SMIT, or the **geninstall** command to install and uninstall these types of packages. The **geninstall** command can detect the format type of a specified package and run the appropriate installation command.

The AIX product media contains **installp** packages and RPM packages that are installed during a base operating system (BOS) installation. The **installp** packages are located in the following path:

```
/mount_point/installp/ppc
```

The RPM packages are located in the following path:

```
/mount_point/RPMS/ppc
```

If you have interim fix packages for AIX, they may be placed in the following path:

```
/mount_point/emgr/ppc
```

If you have media that contains **ISMP** packages for AIX, the **ISMP** packages are located in the following path:

```
/mount_point/ISMP/ppc
```

The **geninstall** command recognizes the following file names as ISMP install images:

- `setupaix*`
- `install*`
- `setup.jar`

If you are using the **geninstall** command to install RPM, interim fix, or **ISMP** packages, use the prefix type to indicate to the **geninstall** command the type of package that you are installing. The package prefix types are the following:

I: **installp** format

R: **RPM** format

J: **ISMP** format

E: **interim fix** format

For example, to install the **cdrecord** RPM package and the **bos.games installp** package, type the following:

```
# geninstall -d/dev/cd0 R:cdrecord I:bos.games
```

The **geninstall** command detects that the **cdrecord** package is an RPM package type and runs the **rpm** command to install the **cdrecord** package. The **geninstall** command then detects that **bos.games** is an **installp** package type and runs the **installp** command to install the **bos.games** package. The process for uninstallation is similar to the installation process.

In SMIT, if you are selecting the packages from a software list, you need not specify the prefix type.

Fileset installation packages

The installation packaging of each fileset in a product can be divided into three parts.

These parts include the usr, root, and share parts. Although this can add further complexity to the understanding of the packaging, this parceling of a software product is necessary for the product to be used by diskless and dataless clients in AIX.

Because they are parceled, a product can be installed on one machine (called the *server*) and then be used remotely by other machines on a network (called the *clients*).

Note: The usr and root parts of a product are packaged in the same installable package.

Item	Description
usr part	The usr part of a software product contains the part of the product that can be shared by machines that have the same hardware architecture. Most of the software that is part of a product usually falls into this category. In a standard system, the usr parts of products are stored in the `/usr` file tree. For example, the **ls** command would be in the `/usr/bin/ls` file.
root part	Every product has a usr part. The root part of a software product contains the part of the product that cannot be shared. The root part of a product is optional because many products may not have any files that need to be specific to each individual machine. In a client/server environment, these are the files for which there must be a unique copy for each client of a server. Most of the root software is associated with the configuration of the machine or product. In a standard system, the root parts of a product are stored in the root (/) file tree. The `/etc/objrepos` directory contains the root part of an installable software product's vital product data (VPD).
share part	The share part of a software product contains the part of the product that can be shared among machines, even if they have different hardware architectures, which can include nonexecutable text or data files. For example, the share part of a product might contain documentation written in ASCII text or data files containing special fonts. The share part of a product is optional because many products might not have any files that can be shared among different hardware platforms. The share part of a product is always packaged in a separately installable package. In a standard system, the share parts of products are usually stored in the `/usr/share` file tree. For example, a dictionary database might be stored in the `/usr/share/dict/words` file.

Creating software packages

The **mkinstallp** command allows users to create their own software packages for AIX.

Packages created with the **mkinstallp** command are in **installp** format and are installed or removed with the **mkinstallp** command.

Files to be packaged by the **mkinstallp** command must be in a directory structure such that the location of the file relative to the root build directory is the same as the destination of the file after installation. For example, if the **/usr/bin/***somecommand* command is to be installed by a **mkinstallp** package, the *somecommand* parameter must be in the *buildroot***/usr/bin** directory when the **mkinstallp** command is invoked.

When the contents of a package are in the correct directory structure, the **mkinstallp** command prompts for basic package data through the command line. This data includes the package name, requisites, descriptions of files to be packaged, and more. The **mkinstallp** command then generates a template file based on responses given by the user. To prevent command line prompting when using a template file, create and edit the template file directly and use the **mkinstallp** command with the **-T** flag.

For example, to package the **/usr/bin/foo** command using the **/tmp/packages** directory as the build root, make sure the following directory structure exists by typing the following at the command line:

```
mkdir /tmp/packages
touch /tmp/packages/usr/bin/foo
```

Then type the following:

```
mkinstallp -d /tmp/packages
```

For more examples, refer to the `/usr/lpp/bos/README.MKINSTALLP` file.

The **mkinstallp** command is included with the **bos.adt.insttools** fileset.

Packaging software bundles

The SMIT Install application look for bundles in **/usr/sys/inst.data/sys_bundles** and in **/usr/sys/inst.data/user_bundles**.

The **sys_bundles** location is typically reserved for system-defined bundles (those which come with AIX). Users can create their own bundle files in the **user_bundles** directory.

The bundle definition file name must end in **.bnd**, because the AIX installation interfaces that process bundles recognize only bundle files that end in .bnd. Use any editor to create bundle files, which can contain comments and fileset names. Lines beginning with the pound sign (#) are recognized as comments and are ignored by the bundle processing code. When you have completed your list of filesets, save the file and make sure the file has the appropriate read permission. Invoking a bundle installation interface displays your bundle without the **.bnd** extension.

The following are examples of the predefined bundles:

- *Server Bundle*. A collection of software packages for machines running AIX in a multiuser standalone or networked environment. This bundle emphasizes functionality over disk utilization.
- *Graphics Bundle*. A collection of software packages that provides support of graphical environments. Graphical support may be automatically installed on some systems during BOS installation.
- *Migration Bundle*. This bundle is created when there was not enough disk space available to complete a migration installation during the BOS installation process. The bundle consists of a collection of software packages that must be installed to complete your migration. You must install this bundle to complete the migration installation. Install the bundle using the **smit update_all** fast path.

  You may also need to install the *Graphics Bundle*.

Some system bundles might refer to installation images that are spread across multiple media. If you see errors indicating that filesets could not be found on the media you are using, insert the media containing the missing filesets and retry the bundle installation.

The system bundles are located in the `/usr/sys/inst.data/sys_bundles` directory. To list the system bundles, type the following:

```
ls /usr/sys/inst.data/sys_bundles/*.bnd
```

You can also use the SMIT `list_bundle` fast path to list the system bundles.

The **geninstall** and **gencopy** commands handle multiple software sources to be specified when a bundle file is used. This is accomplished by grouping software images together under **#MEDIA=** headings in the bundle file. Any images listed under such a heading must reside on the specified media. Media can be specified as the name of a CD (such as *Base Install Media Volume 1* or *AIX Linux Toolbox CD*) or as a local directory (such as the `/usr/sys/inst.images` directory).

The **#MEDIA=** heading is used to designate the location of the file sets or packages in the bundle. For example, the *BaseAndLinuxCD Bundle* might contain the following information:

```
# BaseAndLinuxCDBundle contains packages on volume 1 of base media and on the AIX
# Linux Toolbox CD

#MEDIA=Base Install Media Volume 1
I:bos.adt.prof

#MEDIA=AIX Linux Toolbox CD
R:mtools
R:vim-common
```

When the **geninstall** and **gencopy** commands prompt for the additional media, they use the words provided in the **#MEDIA=** line. In the previous examples, the **geninstall** and the **gencopy** commands

display a message informing you that the **bos.adt.prof installp** package is located on *Base Install Media Volume 1*, and the **mtools** and **vim-common** RPM packages are located on the *AIX Linux Toolbox CD*.

The **#MEDIA=** heading can also be used to indicate a directory. For example, the *CD_Directory Bundle* might contain the following information:

```
# CD_DirectoryBundle contains packages on volume 1 of base install media
# and in /usr/sys/inst.images

#MEDIA=/usr/sys/inst.images
I:bos.games

#MEDIA=Base Install Media Volume 1
I:bos.adt.prof
R:cdrecord-1.9-4
```

This informs the **geninstall** and the **gencopy** commands that the **bos.games installp** package is located in the/usr/sys/inst.images directory, and that the **bos.adt.prof installp** package and the **cdrecord-1.9-4** RPM package are located on *Base Install Media Volume 1*.

The **geninstall** and **gencopy** commands understand the "%L" wildcard in a bundle file. This wildcard is replaced at runtime with the value of the appropriate locale environment variable; **LC_ALL** is checked first, then **LC_MESSAGES**, and then **LANG**. This allows you to create a single bundle file corresponding to multiple installation configurations.

As an example, assume you provide the ABC product, which requires the **abc.rte** and **abc.com** filesets, as well as a message catalog fileset and a documentation fileset. You then provide the message and documentation filesets in English, French, and German, as follows:

```
abc.cat.en_US
abc.cat.fr_FR
abc.cat.de_DE

abc.doc.en_US
abc.doc.fr_FR
abc.doc.de_DE
```

The following bundle file would cause the appropriate combination of filesets to be installed, according to the locale variables on the target system:

```
I:abc.rte
I:abc.com
I:abc.cat.%L
I:abc.doc.%L
```

Note: If expanding the %L wildcard does not yield a fileset name corresponding to a fileset available on the installation media, then the UTF-8 version of the current locale will be tried, then **en_US**, and then **EN_US**.

You can determine whether or not the contents of a bundle are installed on your system with the **lslpp -Lb** command. For example, to determine whether the components of the Alt_Disk_Install bundle, which is located in the /usr/sys/inst.data/sys_bundles directory, are installed, run the following command:

```
lslpp -Lb Alt_Disk_Install.bnd
```

You might see output that looks similar to the following:

```
Fileset                           Level      State  Type  Description
-----------------------------------------------------------------------------------------
bos.alt_disk_install.boot_images  7.1.00     C      F     Alternate Disk Installation Disk Boot Image
bos.alt_disk_install.rte          7.1.00     C      F     Alternate disk Installation Runtime

State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 E -- EFIX Locked.
 O -- Obsolete. (partially migrated to newer version)
 ? -- Inconsistent State...Run lppchk -v.

Type codes:
 F -- Installp Fileset
 P -- Product
 C -- Component
 T -- Feature
 R -- RPM Package
```

Migrating AIX

During a migration, the installation process determines which optional software products are installed on the existing version of the operating system. Components from previous releases are replaced by new software in AIX Version 7.2 are installed at the AIX 7.2 level.

Migration is the default installation method to move from one version and release of AIX to another version and release of AIX, such as from AIX 7.1 and to AIX 7.2.

Beginning with AIX 7.2, the Trusted Computing Base security model is not supported. Therefore, it is disabled during the traditional migration to AIX 7.2 and when you migrate to AIX 7.2 by using the **nimadm** command.

Avoid down-leveling your system when you migrate between different technology levels. For example, when you migrate from AIX 5.3 or AIX Version 6.1 to AIX Version 7.1, you can avoid the risk of down-leveling fixes previously installed on your AIX 5.3 or AIX Version 6.1 system by migrating to the latest available Technology Level of AIX Version 7.1. If you are using a NIM `lpp_source` created with a prior level base media and later levels of updates to be added, create the `lpp_source` with the base media at the same release date or later than the level of AIX 5.3 or AIX Version 6.1 that you are migrating from. The last four digits of the output of the **oslevel -s** command represent the year and week of the service pack currently installed (YYWW). If your system is at a technology level, with no additional service packs installed the last four digits can be "0000". Then, look at the last field from the command "**lslpp -Lc** `bos.sysmgt.sysbr`" to get the same information.

Filesets in AIX 7.2: Beginning with AIX 7.2, to provide you more control over the software that is installed on your system, the `bos.net.tcp.client` and `bos.net.tcp.server` filesets are split into 33 new filesets.

The core code for each original fileset is in the `bos.net.tcp.client_core` and `bos.net.tcp.server_core` filesets. Requisites for software that is shipped with the AIX operating system (the `bos.net.tcp.client` and `bos.net.tcp.server` filesets) are changed to the `bos.net.tcp.client_core` and `bos.net.tcp.server_core` filesets. Additional requisites are added to the other new filesets.

The original filesets still exist to satisfy any requisites from other software. The original filesets have requisites to all the new filesets to ensure that all the requirements are met.

To remove any of these new filesets, you must remove either the `bos.net.tcp.client` fileset or the `bos.net.tcp.server` fileset. To determine the original fileset that has the new fileset, run the **lslpp -d**

new_fileset_name command and look for either the bos.net.tcp.client or bos.net.tcp.server occurrences. If there are no other software that have requisites to the fileset that you want to remove, the removal is possible.

During an operating system migration, code changes occur, so that all the system configuration and user configurable files, which were owned by the bos.net.tcp.client and bos.net.tcp.server filesets, are merged by the new filesets that now own the files.

The list of new filesets follow:

- bos.net.tcp.client_core
- bos.net.tcp.server_core
- bos.net.tcp.bind
- bos.net.tcp.bind_utils
- bos.net.tcp.bootp
- bos.net.tcp.dfpd
- bos.net.tcp.dhcp
- bos.net.tcp.dhcpd
- bos.net.tcp.ftp
- bos.net.tcp.ftpd
- bos.net.tcp.gated
- bos.net.tcp.imapd
- bos.net.tcp.mail_utils
- bos.net.tcp.ntp
- bos.net.tcp.ntpd
- bos.net.tcp.pop3d
- bos.net.tcp.pxed
- bos.net.tcp.rcmd
- bos.net.tcp.rcmd_server
- bos.net.tcp.sendmail
- bos.net.tcp.slip
- bos.net.tcp.slp
- bos.net.tcp.snmp
- bos.net.tcp.snmpd
- bos.net.tcp.syslogd
- bos.net.tcp.tcpdump
- bos.net.tcp.telnet
- bos.net.tcp.telnetd
- bos.net.tcp.tftp
- bos.net.tcp.tftpd
- bos.net.tcp.timed
- bos.net.tcp.traceroute
- bos.net.tcp.x500

Notes:

- If you want to install the next recommended maintenance or technology level of the operating system, use the SMIT **update_all** fast path or the **install_all_updates** command to update the filesets currently

installed. For more information about updating to the next recommended maintenance or technology level of AIX, see “Optional products and service updates” on page 335.

- Before you apply a new Technology Level (TL), you must always create a backup and plan on restoring that backup if you need to rollback to your previous level of the installation. You can also use the**alt_disk_install** or**multibos** options as a way to get back to your previous level. Since TL updates cannot be rejected, you must always commit the updates.
- Before you move your system to a new *version.release* of AIX, you must always create a backup and plan on restoring that backup if you need to roll back to your previous level of the installation. You can also use the **nimadm** command to migrate your system to an alternate disk and maintain the original root volume group on the original disk.

Migration attempts to preserve all user configuration, while it is moving the operating system to a new level of software. The following steps are taken to achieve this objective:

- Save configuration files
- Prepare and remove old files
- Restore new files
- Remove unsupported or unnecessary filesets
- Migrate configuration data wherever possible
- Prepare VPD for installation
- Update more filesets

When you perform a software migration, the following occurs:

- All files in the `/usr/lib/drivers`, `/usr/lib/microcode`, `/usr/lib/methods`, and **`/dev`** directories are removed from the system, so software support for device drivers must be reinstalled. Non-device software products and applications remain on the system, and work correctly if they are among those files that are described in “AIX binary compatibility” on page 408.
- All files in the `/tmp` directory are removed from the system.
- Run the `pre_migration` script for a complete list of software that is removed from your system when you migrate to AIX 7.2. Some of the products are as follow:
  - Director Platform Agent for IBM Systems Director on AIX
  - Director Common Agent
  - Common Agent Services Agent (`cas.agent`)
  - Lightweight Infrastructure Runtime (`lwi.runtime`)
  - System P console (`sysmgt.pconsole`)
  - INed editor (`bos.INed`)
  - Solution Install software (`bos.installers`)
  - Performance Workbench (`bos.perf.gtools`)
  - Eclipse Integrated Tool Platform (`eclipse2`)
  - Network Data Administration Facility (`ndaf`)
  - PEX_PHIGS
  - Pegasus CIM Server and `sysmgt.cim` providers

  Note: Java 5 software is also removed, but there is an option to retain it in the **Base Operating System installation** menus and in the `bosinst_data` resources for network migrations.

In most cases, user-configuration files from the previous version of a product are saved when the new version is installed during a migration installation.

Related information:

Transitioning to POWER8

AIX binary compatibility

AIX binary compatibility allows applications that were created on earlier releases or technology levels of AIX to run unchanged and without recompilation on later releases or technology levels of AIX. For example, an application that is created on AIX 5L can be run on AIX Version 7.1, or later.

The ability to run applications that were created on an earlier version of an operating system on a later level of the operating system is known as backward compatibility. Applications must use only portable programming techniques for binary compatibility on any platform.

The following information describes the application binary compatibility for applications that were created on a specific version of AIX.

Applications from AIX 5L and AIX Version 6.1
: 32-bit and 64-bit applications can run on AIX Version 7.2, or later, without recompilation if the applications use portable programming techniques.

32-bit applications from AIX Version 4
: These applications can run on AIX Version 7.2, or later, without recompilation if the applications use portable programming techniques.

64-bit applications from AIX Version 4
: Any 64-bit applications that are compiled on AIX Version 4 are not binary compatible with AIX 5L, AIX Version 6.1, AIX Version 7.1, or AIX Version 7.2, or later. These versions of AIX are source compatible with 64-bit applications that are created on AIX Version 4. To make 64-bit applications from AIX Version 4 compatible with later version of AIX, you must recompile the application on a system that is running AIX 5L, AIX Version 6.1, AIX Version 7.1, or AIX Version 7.2, or later.

32-bit applications from AIX Version 3
: These applications can run on AIX Version 7.2, or later, without recompilation if the applications use portable programming techniques.

A system that uses AIX Version 7.2, or later, might operate as a server for client machines that are running an earlier version of AIX. In this case, the server operates only if the necessary compatibility options are installed. All conditions about binary compatibility apply in this scenario.

Note: If applications are not running correctly after you migrated to a newer version of the AIX operating system, you can open a Problem Management Report (PMR). When you open the PMR, you must specify "AIX Binary Compatibility" as the subject.

Restrictions for AIX binary compatibility

Applications must use only portable programming techniques for binary compatibility on any platform.

If you create the binary code on a release of the AIX operation system, you cannot run the same binary code on an earlier version of the AIX operation system. For example, if you created the binary code on AIX Version 7.1, you cannot run the same binary code on AIX Version 6.1.

Non-portable programming techniques: The following examples of non-portable programming techniques might affect binary compatibility:

- Applications that are compiled by using a processor-specific compiler option but are run on models other than that processor
- Legacy security library interfaces in which long user names are enabled
- Non-shared compiles of AIX-shared libraries
- X11R5 Server Extensions
- Locales based on IBM-850 code sets

Applications with long user names enabled: AIX Version 5.3, AIX Version 6.1, AIX Version 7.1, and AIX Version 7.2 can be configured to accommodate user names and group names that exceed 8 characters. These versions of the AIX operating system should not be configured for long user names if the systems are running applications that use security library interfaces unless the applications have been tested for long user name support.

Applications might not work correctly on systems that are enabled for long user names and long group names under the following conditions:

- Applications that are not specifically structured to handle long user and group names.
- Applications that use legacy security library interfaces with 8-character name limits.
- Applications that depend on user names and group names not exceeding 8 characters in length.

The following table displays legacy security library interfaces and user name enabled alternatives:

Table 20. User name enabled alternatives

Legacy security library interface long	User name enabled alternative
ckuserID()	authenticatex()
cuserid()	getpwuid()
getuinfo()	getuinfox()
getuinfo_r()	getuinfox()
getuserpw()	getuserpwx()
newpass()	newpassx()
putuserpw()	putuserpwx()
putuserwhist()	putuserpwxhist()

X11R5/X11R6 compatibility issues on AIX Version 7.2: The AIX Version 7.2 X-server uses the X-Consortium release 6 of X (commonly known as X11R6). The libraries that are included by IBM with X11R6 are backwards compatible and the client applications that access these libraries can be used on AIX Version 4, AIX 5L, AIX Version 6.1, and AIX Version 7.1. On these versions of AIX, IBM also includes X11R3, X11R4, and X11R5 compatibility installation options for maximum flexibility.

Most of X-server applications do not cause any problems. However, a few X-server applications use the loadable extension that is provided by the X-server. New functions can be added to the X-server by using extensions. For each extension operation, part of the extension is loaded into the X-server application before the extension can be run. X11R6 modifies how the extension works in the course of improvements to the X-server. The extension modification must be made compatible with X11R6 to run correctly. All extensions that are provided by IBM are compatible. The following extension examples are not compatible with X11R6:

- Sample extensions that are downloaded from the X-Consortium FTP site
- User developed extension
- Third-party extension

In this scenario, the extension needs to be made compatible with X11R6 before the extension can run correctly. User developed extensions and sample X consortium extensions must be recompiled with the X11R6 environment. For third-party extensions, contact the vendor for a X11R6-compatible update.

If you are using non-IBM display adapters, you might also be using vendor supplied software specific to those devices that uses X11R6 server capabilities. In this scenario, the software must be compatible with X11R6 to operate properly. Contact the vendor of the display adapter for the software.

32-bit device drivers and kernel extensions: In AIX Version 6.1, or later, the AIX operating system simplified the kernel environment by providing only the 64-bit kernel. The AIX operating system

maintains application binary compatibility with previous versions of the AIX operating system, but device drivers and kernel extensions that are only 32-bit are not supported on AIX Version 6.1, AIX Version 7.1, and AIX Version 7.2.

Dual-mode (32-bit/64 bit) kernel extensions that are built on AIX 5L can run only in 64-bit mode on AIX Version 6.1, AIX Version 7.1, and AIX Version 7.2.

BOS pre_migration and post_migration checks

The **pre_migration** and **post_migration** commands perform various system checks to ensure a successful migration installation. Both commands are shipped in the **bos.rte** fileset.

In case the **pre_migration** command does not exist on a level of AIX that you want to check before performing a migration installation, the **pre_migration** command is also located in the **usr/lpp/bos** directory of the media file system. Copy the **pre_migration** command from the **usr/lpp/bos** directory of the new AIX media version you are about to perform the migration.

The output from the **pre_migration** command is saved to the system in the **/home/pre_migration** *date* directory.

The **pre_migration** command performs the following actions:

- List the device filesets being removed.
- List all other filesets being removed.
- List the saved base configuration files that will not be merged.
- List configuration files that will be merged.
- Verify fileset version consistency.
- Create a list of all filesets installed, to be used by the **post_migration** command.
- Check the size and location of the boot logical volume.
- Check the major number for rootvg is 10.
- Check for the missing DB directory for the `bos.net.ipsec.keymgt` fileset.
- Determine if Kerberos is being used.
- Check disk and memory sizes.
- Check the firmware level for the IBM Power Systems 7025/7026 systems.

  Note: You can obtain the required version of the firmware from the following Web site, by selecting your product:

  http://www-933.ibm.com/support/fixcentral/

  Refer to the history section of the firmware level for the statement of AIX Version 7.2 support.
- Check whether a standby BOS on the system (as created by the **multibos** command) exists, and if so, it must be removed.
- Check whether the standby BOS has the naming convention of `hd*` for the logical volumes. If yes, the standby BOS will be the group of logical volumes that must be migrated, unless it is removed.
- Check that the level of AIX in the disk control block matches the version and release on the system. If not, the command prompts for appropriate actions.
- If the `bos_hd5` logical volume is the boot logical volume, check whether the `bos_hd4` and `bos_hd2` logical volumes exist.
- If you are migrating from an earlier version of AIX, verify that the correct updates are applied.
- Verify system platform.

Note: If the platform is not supported for AIX Version 7.2, a minimal pre-migration check is performed since the system might be used for a `mksysb` migration.

- Print a recommendation that a system backup be made before the migration.

The output from the **post_migration** command is saved in the **/home/post_migration** *date* directory.

The **post_migration** command performs the following actions:

- Verify fileset version consistency.
- Check the installation list from before the migration, and inform the user of any filesets that might still need migrating.
- Compare saved and merged configuration scripts and save the differences.

Migrating to AIX Version 7.2

Follow this procedure to migrate to AIX Version 7.2.

Notes:

1. The boot logical volume requires 20 MB of contiguous disk space. During migrations, the **inuextendblv** command runs to ensure that there are contiguous partitions for `hd5`. If contiguous partitions are not present, the **inuextendblv** command attempts to create them. If the partitions are not present and the **inuextendblv** command fails to create them, the migration is stopped.
2. The settings in your `bootlist` are not migrated. After a migration, the `bootlist` is set to the primary boot device.

Attention: This procedure requires shutting down and reinstalling the base operating system. Whenever you reinstall any operating system, schedule your downtime when it least impacts your workload to protect yourself from a possible loss of data or functionality. Before you perform a migration installation, ensure that you have reliable backups of your data and any customized applications or volume groups. For instructions on how to create a system backup, refer to "Creating system backups" on page 318.

When you migrate to a newer version of the AIX operating system, know that the default value of the **j2_inodeCacheSize** tunable parameter was changed from 400 to 200. The **j2_inodeCacheSize** tunable parameter allows approximately 50,000 open files per gigabyte (GB) of main memory, and improves system performance. However, the **j2_inodeCacheSize** tunable parameter value of 200 can cause issues in systems that have a small amount of main memory (4 GB or less) and many concurrent users or many concurrent open files. To fix these issues, you can change the values for the **j2_inodeCacheSize** and the **j2_metadataCacheSize** tunable parameters from 200 to the previous value of 400 by running the following command:

Note: When you run the following command, the current value and boot value of both the tunable parameters are reset.

```
ioo -p -o j2_inodeCacheSize=400 -o j2_metadataCacheSize=400
```

If the issues are not fixed after you change the values for the **j2_inodeCacheSize** and the **j2_metadataCacheSize** tunable parameters, you can contact IBM Support.

Prerequisites

Before starting the migration, complete the following prerequisites:

- All requisite hardware, including any external devices (such as tape, CD, or DVD-ROM drives), must be physically connected. If you need further information, refer to the hardware documentation that accompanied your system.
- Before migrating your BOS to AIX 7.2, ensure that the root user has a primary authentication method of **SYSTEM**. You can check this condition by typing the following command:

```
# lsuser -a auth1 root
```

Change the value, if needed, by typing the following command:

```
# chuser auth1=SYSTEM root
```

- Before you begin the installation, other users who have access to your system must be logged off.
- Verify that your applications run on AIX 7.2. Also, verify that your applications are binary-compatible with AIX 7.2. If your system is an application server, verify that there are no licensing issues. Refer to your application documentation or provider to verify on which levels of AIX your applications are supported and licensed.
- Verify that your hardware microcode is up-to-date.
- All requisite hardware, including any external devices (such as tape, CD, or DVD-ROM drives), must be physically connected and powered on. If you need further information, refer to the hardware documentation that accompanied your system.
- Use the **errpt** command to generate an error report from entries in the system error log. To display a complete detailed report, type the following:

  ```
  # errpt -a
  ```

- Adequate disk space and memory must be available. You need at least 4 GB of memory a minimum of 20 GB of physical disk space. See the Disk Requirements topic in the Release Notes, and compare these requirements to the disk usage of the AIX 7.1 system.

  If new file systems are listed for AIX 7.2 that are not on the AIX 7.1 system, these file systems are created during migration. Ensure that the appropriate disk space is available before you begin the migration installation. Migration also takes more disk space for software installations than an overwrite installation.
- Run the pre_migration script.
- Make a backup copy of your system software and data. For instructions, see "Creating system backups" on page 318.

Note: For more information on the supported hardware with AIX 7.2, refer to the Release Notes.

To verify the processor capability, run the following command:

```
/usr/sbin/prtconf -c
```

The command returns `CPU Type: 32-bit` or `CPU Type: 64-bit` depending on the system capability. If your system does not have the **prtconf** command, you can use the **bootinfo -y** command.

Related information:

AIX Release Notes

Preparing your system for BOS installation

Follow this procedure for preparing to migrate to the AIX Version 7.2 BOS.

Prepare for migrating to the AIX Version 7.2 BOS by performing the following steps:

1. Insert the *AIX Volume 1* disk into the media device.
2. Shut down your system. If your machine is currently running, power it off now by following these steps:
   a. Log in as the root user.
   b. Type the following command: **# shutdown -F**
   c. If your system does not automatically power off, place the power switch in the Off (`0`) position.

   Attention: You *must not* turn on the system unit until instructed to do so in "Booting from your installation media" on page 413.
3. Turn on all attached external devices. These include the following:
   - Terminals

- CD or DVD-ROM drives
- Tape drives
- Monitors
- External disk drives

Turning on the external devices first is necessary so the system unit can identify each peripheral device during the startup (boot) process.

Booting from your installation media

Follow this procedure to migrate your current version of the operating system to AIX Version 7.2.

If you are using an ASCII console that was not defined in your previous system, complete "Step 3. Setting up an ASCII terminal" on page 58 before proceeding.

The following steps migrate your current version of the operating system to AIX 7.2:

1. Turn the system unit power switch from Off (`0`) to On (`|`).
2. When the system beeps twice, press F5 on the keyboard (or `5` on an ASCII terminal). If you have a graphics display, you see the keyboard icon on the screen when the beeps occur. If you have an ASCII terminal (also called a tty terminal), you see the word `keyboard` when the beeps occur.

   Note: If your system does not boot using the F5 key (or the `5` key on an ASCII terminal), refer to your hardware documentation for information about how to boot your system from an AIX product media.

   The system begins booting from the installation media.
3. If your system has an LED display, the three-digit LED should display `c31`.

   If you have more than one console, each might display a window that directs you to press a key to identify your system console. A different key is specified for each console displaying this window. If this window displays, press the specified key *only* on the device to be used as the system console. (The system console is the keyboard and display device used for installation and system administration.) Press a key on one console *only*.

   A window displays, asking you to select a language to be used for installation instructions.
4. Select the language you prefer to use for installation instructions.
5. When the Welcome to Base Operating System Installation and Maintenance window displays, either begin the migration immediately by typing `1` to select **Start Install Now with Default Settings**, or verify the installation and system settings by typing `2` to select **Change/Show Installation Settings and Install**. If you want to change any settings, follow the procedure in "Step 5. Verifying or changing the installation settings" on page 59.

   Note:
   - You should not have to change settings simply to select the migration installation method. If a previous version of the operating system exists, the installation method defaults to migration.
   - The available installation methods vary, depending on the version of the operating system that is currently installed (before migration). For information about the BOS installation methods, refer to "Installing the Base Operating System" on page 39. For information about the installation options available for a migration installations, refer to "BOS installation options" on page 44.
6. Verify the selections in the Migration Installation Summary window and press Enter.
7. Confirm the migration installation process in the Migration Confirmation window, and press Enter to begin the migration installation.

Finishing the BOS migration

After prompting for confirmation, the installation process begins. The Installing Base Operating System window displays.

As the installation progresses, the numbers increment in the fields that show percentage complete and elapsed time to indicate the installation status. After the base run-time environment is installed, status information displays about other software that is being installed. After the BOS installation is complete, the system automatically reboots.

After the system has restarted, you are prompted to configure your installation of the BOS. Go to "Configuring AIX" on page 85 for information on the configuration process.

Note: If there is not enough space to migrate all of the usually migrated software, a collection of software called a Migration Bundle is available when you install additional software later. You must create additional disk space on the machine on which you want to install, and then you can run **smit update_all** to complete the installation, during which the Migration Bundle is installed.

If you are not doing the installation from a graphics console, a Graphics_Startup bundle is created. Refer to "Preparing to install optional software products and service updates" on page 338 for more information about installing software bundles and for information on migrating or installing optional software products. "Maintaining optional software products and service updates" on page 343 describes how to remove software from the system to release disk space.
Run the post-migration script and verify the output files.

Checking modifications to configuration files

The **geninstall** command provides an easy way to see what modifications have been made to the configuration files listed in `/etc/check_config.files`.

When these files have been changed during a **geninstall** installation or update operation, the differences between the old and new files is recorded in the `/var/adm/ras/config.diff` file. If `/etc/check_config.files` requests that the old file be saved, the old file can be found in the `/var/adm/config` directory. The `/etc/check_config.files` file can be edited and used to specify whether old configuration files that have been changed should be saved (indicated by `s`) or deleted (indicated by `d`), and has the following format:

```
d /etc/inittab
```

Migrating a multibos instance of AIX

If you previously ran the **multibos** command to create a standby BOS, and restarted the system so that the standby BOS becomes the active BOS, and then removed the new standby BOS, you are running the AIX operating system in an environment that does not have `hd5`, `hd4`, `hd2`, `hd9var`, and `hd10opt` logical volumes, but instead the `bos_hd5`, `bos_hd4`, `bos_hd2`, `bos_hd9var`, and `bos_hd10opt` logical volumes exist. Your system is still recognized as a root volume group (rootvg) during an operating system installation, and the logical volume names are changed to their original names during the migration (or preservation) installation. If you use network alternate disk migration (**nimadm** command) to perform the migration, the logical volume names are changed when you boot the `altinst_rootvg` volume group created by the **nimadm** process for the first time.

If you are running the operating system that has the `bos_*` logical volumes (that is, the **bootinfo -v** command returns `bos_hd5`), but also have a standby instance that has the original `hd*` logical volume names, the standby instance is treated as the rootvg during a migration or preservation installation, and the `bos_*` logical volumes remain unchanged. If you want to migrate the instance that has `bos_*` logical volumes, remove the standby BOS by using the **multibos -RX** command.

You must always back up your system before you migrate an operating system. Before you migrate, copy the `usr/lpp/bos/pre_migration` file from the media or from your network installation manager (NIM) Shared Product Object Tree (SPOT) of the level to which you are migrating, to a location on the target system. Run the file on the target system to check for any migration warnings.

Note: Before you perform a migration or a preservation type of operating system installation in this environment, verify that the disk control block has a valid level for your rootvg. You can run the **/usr/lpp/bosinst/blvset -d /dev/hdiskN -g level** command, where *hdiskN* is the disk that contains the `bos_hd5` logical volume. If this command returns 0.0, run the **bosboot -ad /dev/ipldevice** command to correct it, and run the **blvset** command again to verify the rootvg level. The command must return 6.1 or 7.1.

If you have both standby and active BOS on the system, remove the standby BOS. The originally created `hd*` logical volumes are treated as the operating system, whether it is active or not.

mksysb migration

A **mksysb** migration allows you to restore the **mksysb** from an old system to a system that supports AIX Version 7.2 and then migrate the **mksysb**.

Traditional migration moves the operating system of a supported hardware configuration to a newer level. The **mksysb** migration installation is the recommended method of installation to move unsupported hardware configurations running AIX Version 6.1 and later to new supported hardware running AIX Version 7.2.

A **mksysb** migration is not intended for systems that you can migrate with a traditional migration. This method allows you to bypass the hardware limitation by restoring the **mksysb** on the new hardware configuration and migrate it without running AIX Version 7.2. The resulting system will be running the new level of AIX.

Requirements for using a customized bosinst.data file with a mksysb migration

A customized `bosinst.data` file is required to perform a **mksysb** migration installation.

Your customized `bosinst.data` file must meet the following requirements to be used with a **mksysb** migration:

- The file must be provided using the supplementary diskette method or using the client file method (NIM). For additional information about creating this file, see “Customizing your installation” on page 81. The supplementary CD or DVD method is not supported for a **mksysb** migration.
- The file must contain a new variable called MKSYSB_MIGRATION_DEVICE. This variable specifies the name of the device that contains the **mksysb**. For information about the supported values for this variable, see “bosinst.data control_flow stanza descriptions” on page 46.
- The following variables in the CONTROL_FLOW stanza must be set as follows:
  - PROMPT must be set to *no*.
  - INSTALL_METHOD must be set to *migrate*.
  - EXISTING_SYSTEM_OVERWRITE must be set to *yes*.
  - RECOVER_DEVICES must be set to *no*. A **mksysb** migration attempts to recover the sys0 attributed for the source system as specified in the **mksysb** ODM, but no other device-specific data is recovered from the source system.

  Any user-supplied values for these variable are ignored.
- The file should list the disks to be installed in the TARGET_DISK_DATA stanza to ensure that only those disks are used. A **mksysb** migration is a combination of an overwrite installation and a migration installation. The overwrite portion destroys all of the data on the target disks. The TARGET_DISK_DATA stanza must have enough information to clearly single out a disk. If you supply an empty TARGET_DISK_DATA stanza, the default disk for the platform is used, if available. The following examples show possible values for the TARGET_DISK_DATA stanza:

  Example 1. Disk names only (two disks)

```
target_disk_data:
            PVID =
            PHYSICAL_LOCATION =
```

```
            CONNECTION =
            LOCATION =
            SIZE_MB =
            HDISKNAME = hdisk0

target_disk_data:
            PVID =
            PHYSICAL_LOCATION =
            CONNECTION =
            LOCATION =
            SIZE_MB =
            HDISKNAME = hdisk1
```

Example 2. Physical location specified (1 disk)

```
target_disk_data:
            PVID =
            PHYSICAL_LOCATION = U0.1-P2/Z1-A8
            CONNECTION =
            LOCATION =
            SIZE_MB =
            HDISKNAME =
```

Example 3. By physical volume ID (PVID)(2 disks)

```
target_disk_data:
            PVID = 0007245fc49bfe3e
            PHYSICAL_LOCATION =
            CONNECTION =
            LOCATION =
            SIZE_MB =
            HDISKNAME =

target_disk_data:
            PVID = 00000000a472476f
            PHYSICAL_LOCATION =
            CONNECTION =
            LOCATION =
            SIZE_MB =
            HDISKNAME =
```

Performing the mksysb migration with a DVD installation

You can perform the **mksysb** migration with a DVD installation of AIX Version 7.2.

Prerequisites

- All requisite hardware, including any external devices (such as DVD-ROM drives), must be physically connected. For more information about connecting external devices, see the hardware documentation that accompanied your system.
- Before you begin the installation, other users who have access to your system must be logged off.
- Verify that your applications run on AIX Version 7.2. Also, verify that your applications are binary-compatible with AIX Version 7.2. If your system is an application server, verify that there are no licensing issues. Refer to your application documentation or provider to verify on which levels of AIX your applications are supported and licensed.
- Verify that your hardware microcode is up to date.
- There must be adequate disk space and memory available. AIX Version 7.2 requires minimum of 4 GB of memory and 20 GB of physical disk space. For additional release information, see the *AIX Release Notes*.
- Make a backup copy of your system software and data. For instructions on how to create a system backup, refer to “Creating system backups” on page 318. This backup is used during the **mksysb** migration installation to restore your system files prior to migration.
- If the source system is available, run the pre-migration script on it. Ignore any messages that pertain to the hardware configuration of the source system because the migration takes place on the target system. Correct any other problems as recommended by the script.

Step 1. Prepare your system for installation

Prepare for migrating to the AIX Version 7.2 BOS by completing the following steps:

1. Insert the *AIX Volume 1* disk into the media device.
2. Shut down the target system. If your machine is currently running, power it off by following these steps:
   a. Log in as the root user.
   b. Type `shutdown -F`.
   c. If your system does not automatically power off, place the power switch in the Off (0) position.

   Attention: You must not turn on the system unit until instructed to do so.
3. Turn on all attached external devices. External devices include the following:
   - Terminals
   - CD-ROM drives
   - DVD-ROM drives
   - Tape drives
   - Monitors
   - External disk drives

   Turning on the external devices first is necessary so that the system unit can identify each peripheral device during the startup (boot) process.
4. If your MKSYSB_MIGRATION_DEVICE is a tape, insert the tape for the **mksysb** in the tape drive. If your MKSYSB_MIGRATION_DEVICE is a DVD, and there is an additional DVD drive on the system (other than the one being used to boot AIX), insert the **mksysb** DVD in the drive to avoid being prompted to swap medias.
5. Insert your customized `bosinst.data` supplemental diskette in the diskette drive. If the system does not have a diskette drive, use the network installation method for **mksysb** migration.

Step 2. Boot from your installation media

The following steps migrate your current version of the operating system to AIX Version 7.2. If you are using an ASCII console that was not defined in your previous system, you must define it. For more information about defining ASCII consoles, see "Step 3. Setting up an ASCII terminal" on page 58.

1. Turn the system unit power switch from Off (0) to On (|).
2. When the system beeps twice, press F5 on the keyboard (or 5 on an ASCII terminal). If you have a graphics display, you will see the keyboard icon on the screen when the beeps occur. If you have an ASCII terminal (also called a tty terminal), you will see the word "keyboard" when the beeps occur.

   Note: If your system does not boot using the F5 key (or the 5 key on an ASCII terminal), refer to your hardware documentation for information about how to boot your system from an AIX product media.

   The system begins booting from the installation media. The **mksysb** migration installation proceeds as an unattended installation (non-prompted) unless the MKSYSB_MIGRATION_DEVICE is the same DVD drive as the one being used to boot and install the system. In this case, the user is prompted to switch the product media for the **mksysb** DVD to restore the `image.data` and the `/etc/filesystems` file. After this happens the user is prompted to reinsert the product media and the installation continues. When it is time to restore the **mksysb** image, the same procedure repeats.

The BOS menus do not currently support **mksysb** migration, so they cannot be loaded. In a traditional migration, if there are errors that can be fixed by prompting the user for information through the menus, the BOS menus are loaded. If such errors or problems are encountered during **mksysb** migration, the installation asserts and an error stating that the migration cannot continue displays. Depending on the

error that caused the assertion, information specific to the error might be displayed. If the installation asserts, the LED shows "088".

Step 3. Finish the BOS migration

After the installation process begins, the Installing Base Operating System screen displays.

As the installation progresses, the numbers in the percentage complete field and the elapsed time field increment to indicate the installation status. After the **mksysb** is restored, the base run-time environment is installed, status information about other software that is being installed displays. After the BOS installation is complete, the system automatically reboots.

After the system has restarted, you are prompted to configure your installation of the BOS. For more information on configuring the BOS, see “Configuring AIX” on page 85.

Note:

If there is not enough space to migrate all of the usually migrated software, a collection of software called a migration bundle is available when you install additional software later. You must create additional disk space on the machine where you want to install the migration bundle, and then you can run **smit update_all** to complete the installation where the migration bundle is installed.

If you are not doing the installation from a graphics console, a `Graphics_Startup` bundle is created. For more information on this, see “Optional products and service updates” on page 335. For information on how to remove software from the system to release disk space, see “Maintaining optional software products and service updates” on page 343.

If the pre-migration script ran on the source system, run the post-migration script and verify the output files.

Performing a mksysb migration with NIM installation

You can perform a **mksysb** migration with a NIM installation of AIX Version 7.2.

Prerequisites

- All requisite hardware, including any external devices (such as DVD-ROM drives), must be physically connected. For more information about connecting external devices, see the hardware documentation that accompanied your system.
- Before you begin the installation, other users who have access to your system must be logged off.
- Verify that your applications run on AIX Version 7.2. Also, verify that your applications are binary-compatible with AIX Version 7.2. If your system is an application server, verify that there are no licensing issues. Refer to your application documentation or provider to verify on which levels of AIX Version 7.2 your applications are supported and licensed.
- Verify that your hardware microcode is up-to-date.
- There must be adequate disk space and memory available. AIX Version 7.2 requires minimum 4 GB of memory and 20 GB of physical disk space. For additional release information, see the *AIX Release Notes*.
- Make a backup copy of your system software and data. For instructions on how to create a system backup, refer to “Creating system backups” on page 318. This backup is used during the **mksysb** migration installation to restore your system files prior to migration.
- If the source system is available, run the pre_migration script on it. Ignore any messages that pertain to the hardware configuration of the source system because the migration takes place on the target system. Correct any other problems as recommended by the script.

Step 1. Prepare your system for installation

To prepare your system, verify that the following conditions are met:

- The target system must be a defined client to the NIM master.
- The required customized `bosinst.data` file described in the prerequisites is a NIM `bosinst.data` resource or supplied using the supplemental diskette method.

To instruct the NIM master to start an installation of the client run the following command:

```
# nim -o bos_inst -a source=rte -a spot=spot name -a lpp_source=lpp source name
 -a bosinst_data=bosinst_data resource name -a mksysb=mksysb name client_name
```

The `SPOT` file and `lpp_source` file must be at the AIX Version 7.2 level.

Alternatively, the `mksysb` can be allocated to the client first using a separate **alloc** operation. Then use command line or **smitty nim** to perform a **bos_inst** operation on the client. If the **mksysb** is allocated to the client prior to the **bos_inst** operation, the specification of the **mksysb** is not required.

Step 2. Boot from your installation media

The following steps migrate your current version of the operating system to AIX Version 7.2. If you are using an ASCII console that was not defined in your previous system, you must define the console. For more information about defining ASCII consoles, see “Step 3. Setting up an ASCII terminal” on page 58.

1. After the network boot image is transferred, the system begins booting using the network resources.
2. The **mksysb** migration installation proceeds as an unattended installation (non-prompted).

The BOS menus do not currently support **mksysb** migration, so they cannot be loaded. In a traditional migration, if there are errors that can be fixed by prompting the user for information through the menus, the BOS menus are loaded. If such errors or problems are encountered during **mksysb** migration, the installation asserts and an error stating that the migration cannot continue displays. Depending on the error that caused the assertion, information specific to the error might be displayed. If the installation asserts, the LED shows "088".

Step 3. Finish the BOS migration

After the installation process begins, the Installing Base Operating System screen displays.

As the installation progresses, the numbers in the percentage complete field and the elapsed time field increment to indicate the installation status. After the **mksysb** is restored, the base run-time environment is installed, status information about other software that is being installed displays. After the BOS installation is complete, the system automatically reboots.

After the system has restarted, you are prompted to configure your installation of the BOS. For more information on configuring the BOS, see “Configuring AIX” on page 85.

Note:

If there is not enough space to migrate all of the usually migrated software, a collection of software called a migration bundle is available when you install additional software later. You must create additional disk space on the machine where you want to install the migration bundle, and then you can run **smit update_all** to complete the installation where the migration bundle is installed.

If you are not doing the installation from a graphics console, a `Graphics_Startup` bundle is created. For more information on this, see “Optional products and service updates” on page 335. For information on how to remove software from the system to release disk space, see “Maintaining optional software products and service updates” on page 343.

If the pre-migration script ran on the source system, run the post-migration script and verify the output files.

Partitioning

Partitioning your system is similar to partitioning a hard disk drive. When you partition a hard disk drive, you divide a single physical hard drive so that the operating system recognizes it as a number of separate logical hard drives.

You have the option of dividing the system's resources by using the Hardware Management Console (HMC) to partition your system. On each of these divisions, called *partitions*, you can install an operating system and use each partition as you would a separate physical system.

Partitioning concepts

Before you can start installing BOS on partitions you need to learn about general ideas and terminology.

Logical partitions

A *logical partition* (LPAR) is the division of a computer's processors, memory, and hardware resources into multiple environments so that each environment can be operated independently with its own operating system and applications.

The number of logical partitions that can be created depends on the system's processor model and resources available. Typically, partitions are used for different purposes, such as database operation, client/server operations, Web server operations, test environments, and production environments. Each partition can communicate with the other partitions as if each partition is a separate machine.

The AIX operating system supports partitioned environments. Although the AIX installation concepts are the same, the configuration and management of a partitioned environment with the AIX operating system are new.

A logical partition must contain a minimum set of resources, as follows:

- 1 GB of available system memory
- One available system processor
- One boot device on an assigned I/O slot
- One available network adapter (for error reporting)
- Any other adapters you might need on an assigned I/O slot

Processors, memory, and I/O slots can be allocated to any partition, regardless of their location. However, if you attempt to activate a partition, but the resources you specified are not available at the time, partition activation fails. It is important to keep track of your system's resources to avoid activation failures. PCI slots are assigned individually to partitions, and memory can be allocated in 256 MB increments. The granularity of the resources that can be assigned to partitions is very fine, providing flexibility to create systems with just the desired amount of resources. Each partition runs its own copy of the AIX operating system and is isolated from any activity in other partitions. Software failures do not propagate through the system, and the hardware facilities and microcode isolate the resources.

Managed system

A managed system is a system that is physically attached to and managed by the Hardware Management Console (HMC).

You can use the HMC to perform tasks that affect the entire managed system, such as powering the system on and off. You can also create partitions and partition profiles within each managed system. These partitions and partition profiles define the way that you configure and operate your partitioned system.

Dynamic logical partitioning

Dynamic logical partitioning provides the ability to logically attach and detach a managed system's resources to and from a logical partition's operating system without rebooting.

For more information on dynamic logical partitioning, see the following:

- Dynamic logical partitioning in *Performance management*
- Dynamic logical partitioning in *General Programming Concepts: Writing and Debugging Programs*

Affinity logical partitions

An *affinity logical partition* is a special type of logical partition that uses system resources that are in close physical proximity to each other.

Some systems have the ability to create affinity logical partitions. Check your hardware specifications to see if your managed system is capable of using affinity logical partitions. When creating an affinity logical partition, the HMC automatically determines which system resources are to be used, based on their physical location to each other. The system resources that are automatically managed by the HMC are processors and memory. The user determines the I/O requirements for each of these partitions. The HMC then creates a profile for each affinity logical partition and a system profile for the managed system.

Full system partition

A special partition called the *Full System Partition* assigns all of your managed system's resources to one large partition.

The Full System Partition is similar to the traditional, non-partitioned method of operating a system. Because all resources are assigned to this partition, you cannot start any other partitions when the Full System Partition is running. You also cannot start the Full System Partition when other partitions are running. You should choose to use either the Full System Partition or create other partitions. Your I/O usage might be affected if you switch between these two options frequently.

Running AIX on a logical partition

There are several differences between how AIX runs on a logical partition and how it runs on a standalone server.

The following list describes some of these differences:

- The logical partition resource allocation provides the ability to select individual components to be added to a partition without dependencies between these resources. The slots can be freely allocated in any I/O drawer on the system. Other devices may be required for specific application requirements. It is a good idea to configure more PCI slots in the partition than are required for the number of adapters. This provides flexibility by allowing additional adapters to be hot-plugged into the empty slots that are part of an active partition. Because each partition requires its own separate boot device, the system must have at least one boot device and associated adapter per partition.
- In order for AIX to run inside a logical partition, AIX calls the Hypervisor in place of its traditional direct access to the hardware and address-mapping facilities.
- Some direct-access calls are presented for diagnostic purposes, and alternate return codes for Run-Time Abstraction Services (RTAS) calls are used whenever an illegal operation is issued.
- No physical console exists on the partition. While the physical serial ports on the system can be assigned to the partitions, they can only be in one partition at a time. To provide an output for console messages and also for diagnostic purposes, the firmware implements a virtual tty that is seen by AIX as a standard tty device. Its output is sent to the HMC. The AIX diagnostics subsystems use the virtual tty as the system console.
- Certain platform operations are constrained in LPARs. For example, in non-LPAR systems, platform firmware updates can be performed from AIX by a root user. Because firmware updates can affect all

partitions in an LPAR system, the LPAR administrator can specify that a particular partition (or no partition) has this authority. Within that partition, firmware updates work in the same way as they do for non-LPAR systems.

Apart from these considerations, AIX runs within a partition the same way it runs on a standalone server. No differences are observed either from the application or the administrator's point of view. Third-party applications need only be certified for a level of AIX that runs in a partition, and *not* for the LPAR environment itself.

Remote management

You can connect your browser to the Hardware Management Console (HMC) to manage your partitions remotely or use the command line.

Each of the following system-management methods can be performed using the HMC interface or the command line.

- Use the HMC client to remotely manage any AIX partition or system. All AIX plug-ins on the AIX system can be managed remotely from the HMC client.
- Use an HMC client remotely manage another HMC client. All HMC plug-ins on the HMC server may be managed remotely from the HMC client. The only plug-in that is an exception is the Service Agent plug-in.

For information about remotely managing partitions with the command line on the HMC, see Using the HMC remote command line.

Partition security

System administrators can install a server with the Evaluation Assurance Level 4+ (EAL4+) option during a base operating system (BOS) installation. If you select this option, there are restrictions on the software that is installed during BOS installation and network access restrictions.

Starting with AIX 5L Version 5.2 with the 5200-01 Recommended Maintenance package, the EAL4+ technology runs on POWER4 processor hardware platforms that support logical partition configuration. The following peripherals are supported on EAL4+ servers:

- Storage devices
  - Terminals
  - Printers
  - Hard disks
  - CD-ROM drives
- Backup devices
  - Streamers
  - Floppy disk drives
- Network devices
  - Ethernet
  - Token ring

Implementations of logical partitions

A logically partitioned environment adds to a portfolio of solutions that can provide better management, improved availability, and more efficient use of resources. You can implement logical partitions in multiple ways.

Server consolidation

If you have a server with sufficient processing capacity, you can logically subdivide the server into a number of separate smaller systems to enable server consolidation. Using partitioning for sever

consolidation allows you to isolate applications, with the additional benefits of reduced floor space, a single point of management, and easier redistribution of resources as workloads change.

Mixed production and test environments

Usually, production and test environments should be isolated from each other. Partitioning enables separate partitions to be allocated for the production and test systems, eliminating the need to purchase additional hardware and software.

When testing has been completed, the resources allocated to the test partition can be returned to the production partition or elsewhere as required. You can also add extra resources to a partition if you want to move the partition from a test environment to a production environment. As new projects are developed, they can be built and tested on the same hardware where they will be deployed.

Consolidation of multiple versions of the same operating system

Different versions of AIX can exist on different logical partitions (LPARs) on the same system.

Consolidating multiple versions of AIX on a single system allows you to accommodate multiple application requirements without multiple systems. You can also create an LPAR to test applications under new versions of the operating system before you upgrade the production environments. Instead of having a separate server for this function, a minimum set of resources can be temporarily used to create a new LPAR where you test the application. When you no longer need the partition, you can incorporate its resources back into the other LPARs.

Network adapter communication between partitions and the HMC

After a partition has been started, it uses the network adapter to communicate with the Hardware Management Console (HMC).

Both the HMC and the partition must be configured so they can use the network adapters to communicate with each other. The partition must be configured to identify the HMC (or HMCs) on the network. It is recommended that the network be configured using a Domain Name Service (DNS) server.

You can use either fully qualified host names or short host names to identify partitions and HMCs. However, it is recommended that each partition and HMC be identified using a fully qualified host name, as this identification ensures unique naming of all the partitions and the HMC in the network. Fully qualified host names cannot be more than 100 bytes in length.

The HMC and partitions can also be configured using a short host name, where the domain name is not defined. This is typically done in a private or test network. If the HMC is defined using a short host name, you must perform extra network configuration steps to ensure correct communications between the partitions and the HMC. If you use short host names rather than fully qualified host names, make sure that the short host names are unique and that the mappings to IP addresses are properly specified.

The search order between the local `/etc/hosts` file and the DNS can be specified using the `/etc/netsvc.conf` file or `/etc/irs.conf` file.

The following examples illustrate the scenarios supported:

- If you are using DNS and your partition and the HMC are using fully qualified host names, then no additional network configuration is required.
- If you are using DNS and your partition is using a short host name, such as *partition_1* and the HMC is also using a short host name, such as *hmc123*, both must be added to the local **/etc/hosts** file, as shown in the following:

```
  root@partition_1
-> cat /etc/hosts
```

```
127.0.0.1  loopback localhost

9.3.3.151 partition_1.mydomain.mycompany.com  partition_1
9.3.3.152 hmc123.mydomain.mycompany.com  hmc123
```

Note: You must include the fully qualified host name in addition to the short name when a DNS is present.

- If you are not using DNS and your partition is using a fully qualified host name, such as *partition_1.mydomain.mycompany.com*, and the HMC is also using a fully qualified host name, such as *hmc123.mydomain.mycompany.com*, both must be added to the local `/etc/hosts` file, as shown in the following:

```
  root@partition_1.mydomain.mycompany.com
-> cat /etc/hosts

127.0.0.1  loopback localhost
9.3.3.151  partition_1.mydomain.mycompany.com
9.3.3.152 hmc123.mydomain.mycompany.com
```

- If you are not using DNS and your partition is using a short host name, such as *partition_1* and the HMC is also using a short host name, such as *hmc123*, both must be added to the local `/etc/hosts` file, as shown in the following:

```
  root@partition_1
-> cat /etc/hosts

127.0.0.1  loopback localhost

9.3.3.151 partition_1
9.3.3.152 hmc123
```

- Your HMC is using a short host name, such as *hmc123*, and you would like to use both a fully qualified host name and a short host name for the HMC. In order for your partition to correctly communicate with the HMC, you must specify the short host name before the fully qualified host name in the partition's `/etc/hosts` file, as shown in the following:

```
  root@partition_1.mydomain.mycompany.com
-> cat /etc/hosts

127.0.0.1  loopback localhost

9.3.3.151 partition_1.mydomain.mycompany.com
9.3.3.152 hmc123 hmc123.mydomain.mycompany.com
```

Installing AIX in a partitioned environment

There are multiple procedures for installing AIX in a partitioned environment.

For the installation method that you choose, ensure that you follow the sequence of steps as shown. Within each procedure, you must use AIX to complete some installation steps, while other steps are completed using the HMC interface.

Installing AIX using the media device to install a partition with an HMC

In this procedure, you will perform a new and complete base operating system installation on a logical partition using the partition's media device. This procedure assumes that there is an HMC attached to the managed system.

Prerequisites

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

Note: For the installation method that you choose, ensure that you follow the sequence of steps as shown. Within each procedure, you must use AIX to complete some installation steps, while other steps are completed using the HMC interface.

Before you begin this procedure, you should have already used the HMC to create a partition and partition profile for the client. Assign the SCSI bus controller attached to the media device, a network adapter, and enough disk space for the AIX operating system to the partition. Set the boot mode for this partition to be SMS mode. After you have successfully created the partition and partition profile, leave the partition in the *Ready* state. For instructions about how to create a logical partition and partition profile, refer to the Creating logical partitions and partition profiles article in the IBM Power Systems Hardware Information Center.

Step 1. Activate and install the partition (perform these steps in the HMC interface)

1. Activate the partition, as follows:
   a. Insert the *AIX 7 Volume 1* media into the media device of the managed system.
   b. In the navigation panel, open **Systems Management > Servers**, and click the system on which the logical partition is located.
   c. From the Tasks menu, select partition, click **Operations > Activate > Profile**.
   d. Select **Open a terminal window or console session** at the bottom of the menu to open a virtual terminal (vterm) window.
   e. Select **Advanced** to open the Advanced options menu.
   f. For the Boot mode, select **SMS**.
   g. Select **OK** to close the Advanced options menu.
   h. Select **OK**. A vterm window opens for the partition.
2. In the SMS menu on the vterm, do the following:
   a. Press the 5 key and press Enter to select **5. Select Boot Options**.

```
PowerPC Firmware
Version SF220_001
SMS 1.5 (c) Copyright IBM Corp. 2000, 2003  All rights reserved.
-------------------------------------------------------------------------------
Main Menu

1. Select Language
2. Setup Remote IPL (Initial Program Load)
3. Change SCSI Settings
4. Select Console
5. Select Boot Options


-------------------------------------------------------------------------------
Navigation Keys:

                X = eXit System Management Services
-------------------------------------------------------------------------------
Type the number of the menu item and press Enter or select Navigation Key: 5
```

   b. Press the 2 key and press Enter to select **2. Select Boot Devices**.
   c. Press the 1 key and press Enter to select **1. Select 1st Boot Device**.
   d. Press the 3 key and press Enter to select **3. DVD**.
   e. Select the media type that corresponds to the media device and press Enter.
   f. Select the device number that corresponds to the media device and press Enter. The media device is now the first device in the Current Boot Sequence list.
   g. Press the ESC key until you return to the Configure Boot Device Order menu.
   h. Select the device number that corresponds to the hard disk and press Enter.
   i. Press the x key to exit the SMS menu. Confirm that you want to exit SMS.

3. Boot from the *AIX Volume 1*, as follows:
   a. Select console and press Enter.
   b. Select language for BOS Installation menus, and press Enter to open the Welcome to Base Operating System Installation and Maintenance menu.
   c. Type 2 to select **Change/Show Installation Settings and Install** in the **Choice** field and press Enter.

```
                    Welcome to Base Operating System
                    Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

    1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    88  Help ?
    99  Previous Menu
>>> Choice [1]: 2
```

4. Verify or Change BOS Installation Settings, as follows:
   a. Type 1 in the **Choice** field to select the **System Settings** option.
   b. Type 1 for New and Complete Overwrite in the **Choice** field and press Enter.

      Note: The installation methods available depend on whether your disk has a previous version of AIX installed.
   c. When the Change Disk(s) screen opens, you can change the destination disk for the installation. If the default shown is correct, type 0 in the **Choice** field and press Enter. To change the destination disk, do the following:
      1) Type the number for each disk you choose in the **Choice** field and press Enter. *Do not* press Enter a final time until you have finished selecting all disks. If you must deselect a disk, type its number a second time and press Enter.
      2) When you have finished selecting the disks, type 0 in the **Choice** field and press Enter. The Installation and Settings screen opens with the selected disks listed under **System Settings**.
   d. If needed, change the primary language environment. Use the following steps to change the primary language used by this installation to select the language and cultural convention you want to use.

      Note: Changes to the primary language environment do not take effect until after the Base Operating System Installation has completed and your system is rebooted.
      1) Type 2 in the **Choice** field on the Installation and Settings screen to select the **Primary Language Environment Settings** option.
      2) Select the appropriate set of cultural convention, language, and keyboard options. Most of the options are a predefined combination, however, you can define your own combination of options.
         - To choose a predefined Primary Language Environment, type that number in the **Choice** field and press Enter.
         - To configure your own primary language environment, do the following:
           a) Select **MORE CHOICES**.
           b) Select **Create Your Own Combination**.
           c) When the Set Primary Cultural Convention screen opens, type the number in the **Choice** field that corresponds to the cultural convention of your choice and press Enter.

d) When the Set Primary Language screen opens, type the number in the **Choice** field that corresponds to your choice for the primary language and press Enter.

e) When the Set Keyboard screen opens, type the number in the **Choice** field that corresponds to the keyboard attached to the system and press Enter.

e. After you have made all of your selections, verify that the selections are correct. Press Enter to confirm your selections and to begin the BOS Installation. The system automatically reboots after installation is complete.

5. Switch the partition to Normal Mode, as follows:
   a. Right-click on the partition profile to open the menu. Be sure the correct partition profile is highlighted.
   b. Select **Properties**.
   c. Select the **Settings** tab.
   d. For the Boot Mode, select `Normal`.
   e. Select **OK** to close the Properties menu.
   f. Right-click on the partition to open the menu.
   g. Select **Restart Partition**.
   h. Select **Immediate** for the Restart Options.
   i. Confirm that you want to restart the partition.
   j. When the partition has restarted, right-click on the partition to open the menu.
   k. Select **Open terminal window** to open a virtual terminal (vterm) window.
6. Complete the BOS Installation, as follows:
   a. Type `vt100` as the terminal type.

```
                          Set Terminal Type
The terminal is not properly initialized.  Please enter a terminal type
and press Enter.  Some terminal types are not supported in
non-English languages.

     ibm3101          tvi912          vt330
     ibm3151          tvi920          vt340
     ibm3161          tvi925          wyse30
     ibm3162          tvi950          wyse50
     ibm3163          vs100           wyse60
     ibm3164          vt100           wyse100
     ibmpc            vt320           wyse350
     lft              sun

                     +----------------------Messages-------------------------
                     | If the next screen is unreadable, press Break (Ctrl-c)
    88  Help ?       | to return to this screen.
    99  Exit         |
                     |
>>> Choice []: vt100
```

   b. In the License Agreement menu, select **Accept License Agreements**.
   c. Select **yes** to ACCEPT Installed License Agreements.
   d. Press F10 (or Esc+0) to exit the License Agreement menu.
   e. In the Installation Assistant main menu, select **Set Date and Time**.

```
                          Installation Assistant

Move cursor to desired item and press Enter.

  Set Date and Time
  Set root Password
  Configure Network Communications
  Install Software Applications
  Using SMIT (information only)


F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

f. Set the correct date, time, and time zone. Press the F3 (or Esc+3) key to return to the Installation Assistant main menu.
g. Select **Set root Password**. Set a root password for the partition.
h. Select **Configure Network Communications**. Select **TCP/IP Startup**. Select from the Available Network Interfaces and press Enter. Enter the appropriate network information in the Minimum Configuration and Startup menu and press Enter. Use the F3 (or Esc+3) key to return to the Installation Assistant main menu.
i. Exit the Installation Assistant by pressing F10 (or Esc+0).
j. The vterm window displays a login prompt.

Step 2. Manage your partition (perform this step in the AIX environment)

When the installation has completed and the system has rebooted, the vterm window displays a login prompt.

At this point, you may want to perform several common system-administration procedures. The following table lists where to find information about performing these procedures.

Table 21. **Common System Administration Procedures**

Procedure	Location
Backing up and recovering system backups	"Creating and installing system backups" in *Installation and migration*
Managing users and groups	"Users, Roles, and Passwords" in *Security*
Installing software	"Optional Software Products and Service Updates" in *Installation and migration*
Installing fixes/updates	"Optional Software Products and Service Updates" in *Installation and migration*
Tuning the system for performance	"Performance tuning" in *Performance management*
Configuring printers	*Printers and printing*

Installing AIX using the media device to install a partition without an HMC

In this procedure, you will use the system's built-in media device to perform a new and complete Base Operating System Installation on the standalone system.

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

This information contains procedures to install the AIX operating system. For more information on concepts and considerations involved when performing a base operating system installation of AIX, or concepts and requirements involved when using the Network Installation Manager (NIM) to install and maintain AIX, refer to *Installation and migration*.

At this point, the BOS Installation is complete, and the initial configuration of the system is complete.

Step 1. Prepare your system for installation

- There must be adequate disk space and memory available. AIX requires 4 GB of memory and 20 GB of physical disk space. For additional release information, see the *AIX 7.2 Release Notes*.
- Make sure your hardware installation is complete, including all external devices. See the documentation provided with your system unit for installation instructions.
- If your system needs to communicate with other systems and access their resources, make sure you have the information in the following worksheet before proceeding with installation:

Table 22. Network Configuration Information Worksheet

Network Attribute	Value
Network Interface	(For example: en0, et0)
Host Name	
IP Address	_____._____._____._____
Network Mask	_____._____._____._____
Nameserver	_____._____._____._____
Domain Name	
Gateway	_____._____._____._____

Step 2. Boot from the AIX product media

1. Insert the *AIX Volume 1* media into the media device.
2. Make sure all external devices attached to the system (such as DVD drives, and terminals) are turned on. Only the media drive from which you will install AIX should contain the installation media.
3. Follow whatever procedure is needed to power on the system to cause it to boot from an AIX product media. Consult your hardware documentation for instructions if necessary.

   Note: Most older MicroChannel systems require the keylock to be set in the service position before powering on the system. Some older PCI systems require you to type `5` or press the F5 key (depending on whether you have an ASCII terminal or color graphics display console) when the system beeps and begins repeating `IBM` on the console several seconds after being powered on. Most current PCI systems only require that you repetitively type the `5` key (regardless of what type of console you have) at these system prompts. Also, most current systems can be set to boot from alternate media before they are powered on using the service processor menu. Consult your hardware documentation for more information.
4. Select the system console when prompted by typing the key indicated by the prompt (1, 2, F1, F2, and so on).
5. Select the English language for the base operating system (BOS) Installation menus by typing a `1` in the **Choice** field. Press Enter to open the Welcome to Base Operating System Installation and Maintenance screen.
6. Type `2` to select **2 Change/Show Installation Settings and Install** in the **Choice** field and press Enter.

```
                    Welcome to Base Operating System
                    Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

    1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery

    4 Make Additional Disks Available

    5 Select Storage Adapters

    88  Help ?
    99  Previous Menu
>>> Choice [1]: 2
```

Step 3. Set and verify BOS installation settings

1. In the Installation and Settings screen, verify the installation settings are correct by checking the method of installation (new and complete overwrite), the disk or disks you want to install, the primary language environment settings, and the advanced options.
2. To change the System Settings, which includes the method of installation and disk where you want to install, type 1 in the **Choice** field and press Enter.

```
                        Installation and Settings

Either type 0 and press Enter to install with current settings, or type the
 number of the setting you want to change and press Enter.

    1  System Settings:
         Method of Installation.............New and Complete Overwrite
         Disk Where You Want to Install.....hdisk0

>>> Choice [0]: 1
```

3. Type 1 for New and Complete Overwrite in the **Choice** field and press Enter. The Change Disk(s) Where You Want to Install screen now displays.

```
                Change Disk(s) Where You Want to Install

Type one or more numbers for the disk(s) to be used for installation and press
Enter. To cancel a choice, type the corresponding number and Press Enter.
At least one bootable disk must be selected. The current choice is indicated
by >>>.

        Name      Location Code   Size(MB)  VG Status   Bootable

     1  hdisk0   04-B0-00-2,0      30720   none            Yes
     2  hdisk1   04-B0-00-5,0      30720   none            Yes
     3  hdisk2   04-B0-00-6,0      12288   none            Yes

>>>  0   Continue with choices indicated above

    66  Disks not known to Base Operating System Installation
    77  Display More Disk Information
    88  Help ?
    99  Previous Menu

>>> Choice [0]:
```

4. In the Change Disk(s) Where You Want to Install screen:
   a. Select **hdisk0** by typing a 1 in the **Choice** field and press Enter. The disk will now be selected as indicated by **>>>**. To unselect the destination disk, type the number again and press Enter.

b. To finish selecting disks, type a `0` in the **Choice** field and press Enter. The Installation and Settings screen displays with the selected disks listed under **System Settings**.

5. Change the Primary Language Environment Settings to English (United States). Use the following steps to change the Cultural Convention, Language, and Keyboard to English.
   a. Type `2` in the **Choice** field on the Installation and Settings screen to select the **Primary Language Environment Settings** option.
   b. Type the number corresponding to English (United States) as the Cultural Convention in the **Choice** field and press Enter.
   c. Select the appropriate keyboard and language options.
6. Verify that the selections are correct in the Overwrite Installation Summary screen, as follows:

```
     Overwrite Installation Summary

Disks:  hdisk0
Cultural Convention:  en_US
Language:  en_US
Keyboard:  en_US
Graphics Software:  Yes
Desktop:  CDE
System Management Client Software:  Yes
OpenSSH Client Software:  No
OpenSSH Server Software:  No
Enable System Backups to install any system:  Yes
Selected Edition:  express

Optional Software being installed:

>>> 1   Continue with Install
    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

7. Press Enter to begin the BOS installation. The system automatically reboots after installation is complete.

Step 4. Configure the system after installation

1. On systems with a graphics display, after a new and complete overwrite installation, the Configuration Assistant opens. On systems with an ASCII display, after a new and complete overwrite installation, the Installation Assistant opens.
2. Select the **Accept Licenses** option to accept the electronic licenses for the operating system.
3. Set the date and time, set the password for the administrator (root user), and configure network communications (TCP/IP).

   Use any other options at this time. You can return to the Configuration Assistant or the Installation Assistant by typing `configassist` or `smitty assist` at the command line.
4. Select **Exit the Configuration Assistant** and select **Next**. Or, press F10 (or `ESC+0`) to exit the Installation Assistant.
5. If you are in the Configuration Assistant, select **Finish now, and do not start Configuration Assistant when restarting AIX** and select **Finish**.

Step 5. Manage your system

At this point, you may want to perform several common system-administration procedures. The following table lists where to find information about performing these procedures.

Table 23. **Common System Administration Procedures**

Procedure	Location
Backing up and recovering system backups	"Creating and installing system backups" in *Installation and migration*
Managing users and groups	"Users, Roles, and Passwords" in *Security*
Installing software	"Optional Software Products and Service Updates" in *Installation and migration*
Installing fixes/updates	"Optional Software Products and Service Updates" in *Installation and migration*
Tuning the system for performance	"Performance tuning" in *Performance management*
Configuring printers	*Printers and printing*

Installing a partition using alternate disk installation

You can clone an existing disk image to another disk or disks without using NIM.

You might want to use an alternate disk installation if your network is not fully set up, or if you are not sure about your network configuration. Using an alternate disk installation will not prevent you from using NIM in the future.

You can use the **alt_disk_install** command to clone a system image to another disk, but you must use the **-O** option to remove references in the object data manager (ODM) and device (`/dev`) entries to the existing system. The **-O** flag specifies that the **alt_disk_install** command should call the **devreset** command to reset the device database. The cloned disk can now be booted as if it were a new system.

For a full description of alternate disk installation, see **alt_disk_install**.

1. Boot the managed system as a Full System Partition so you have access to all the disks in the managed system.
2. Configure the system and install the necessary applications.
3. Run the **alt_disk_install** command to begin cloning the `rootvg` on `hdisk0` to `hdisk1`, as follows:

   ```
   # /usr/sbin/alt_disk_install -O -B -C hdisk1
   ```

   The cloned disk (`hdisk1`) will be named `altinst_rootvg` by default.
4. Rename the cloned disk (`hdisk1`) to `alt1` as follows:

   ```
   # /usr/sbin/alt_disk_install -v alt1 hdisk1
   ```

   Renaming the cloned disk allows you torepeat the operation with another disk.
5. Run the **alt_disk_install** command again to clone to another disk and rename the cloned disk, as follows:

   ```
   # /usr/sbin/alt_disk_install -O -B -C hdisk2
   # /usr/sbin/alt_disk_install -v alt2 hdisk2
   ```
6. Repeat steps 3 through 5 for all of the disks that you want to clone.
7. Use the HMC to partition the managed system with the newly cloned disks. Each partition you create will now have a `rootvg` with a boot image.
8. Boot the partition into SMS mode. Use the SMS **MultiBoot** menu to configure the first boot device to be the newly installed disk.
9. Exit the SMS menus and boot the system.

Configuring an initial partition as a NIM master to use NIM to install the remaining partitions

You can set up an initial logical partition as a NIM master and server. The NIM environment allows you to manage installations for your other partitions.

Before you begin this procedure, you should perform the following tasks:

- Use the HMC to create the Master_LPAR partition profile. Leave the partition in the *Ready* state.
- Verify that the Master_LPAR partition has a network adapter, enough hard-disk space for the NIM resources, and an assigned CD device.
- Set the boot mode for the Master_LPAR partition to be Normal mode.
- Use the HMC to create logical partitions and partition profiles for each NIM client.
- Verify that each NIM client partition has a network adapter assigned. Set the boot mode for each partition to be SMS mode. After you have successfully created the partitions and partition profiles, leave the partitions in the *Ready*
- If AIX is not currently installed on any of the disks in the system, you must install it. For more information, see "Installing AIX using the media device to install a partition with an HMC" on page 24.
- Configure AIX for network communication on the Master_LPAR.
- Activate the Master_LPAR partition profile on the HMC.

  You should consider the following aspects of the installation when you configure an initial partition as a NIM Master and using NIM to install the remaining partitions has the following advantages:
  - NIM environment offers the most flexibility and customization options for installation and management.
  - NIM environment allows for multiple installations at the same time.
  - The `nim_master_setup` and `nim_clients_setup` scripts provide a way to set up the NIM environment.
  - Requires one LPAR with approximately 1.5 GB of disk space dedicated as the NIM master partition.

  In this procedure, you will set up an initial logical partition as a NIM master and server. This procedure refers to this initial logical partition as the *Master_LPAR*.

1. Run the **oslevel** command on the Master_LPAR. Output similar to the following displays:
   ```
   5200
   ```
   If the output from the **oslevel** command does not show the expected OS level, see "Migrating AIX" on page 405 for information about migrating the AIX operating system to the correct OS level.
2. Verify your network connection by running the **netstat** command. You can run this command with the **-C** flag to show routing table information. You can also use the **-D** flag to show the number of packets received, transmitted, and dropped in the communications subsystem.
3. Insert the AIX installation media.
4. Run the **nim_master_setup** command. For additional information on options for running this command, see **nim_master_setup**.
5. Run the following command to open the`/export/nim/client.defs` file with the vi editor:
   ```
   # vi /export/nim/client.defs
   ```
6. Edit the `client.defs` file according to your environment. For more information on this file, see the instructions and examples in the `client.defs` file. When you are finished editing the `client.defs` file, save it and exit the vi editor.
7. Run the **nim_clients_setup -c** command. For additional information on options for running this command, see **nim_clients_setup**.

   Note: If you are adding new client machines that cannot be resolved on the name server, edit the `/etc/hosts` file to add the IP addresses and client host names.

After you have defined the client machines, you should activate and install the partitions. For more information on activating partitions, see Activating a partition profile.

After you activate and install the client partitions, you can perform any system management tasks.

Related information:

Activating a partition profile

Using a separate AIX system as a NIM master to use NIM to install each partition

You can use a separate system running AIX as a NIM master and server. The NIM environment allows you to manage installations for your other partitions.

Before you begin this procedure, you should perform the following tasks:

- Use the HMC to create partitions and partition profiles for each NIM client partition that you want to install. Leave the partitions in the *Ready* state.
- Verify each partition has a network adapter assigned.
- Set the boot mode for each partition to SMS mode.

You should consider the following aspects of the installation when you use a separate AIX system as a NIM Master to use NIM to install each partition has the following advantages:

- You need not dedicate an LPAR as the NIM master.
- NIM environment offers the most flexibility and customization options for installation and management.
- NIM environment allows for multiple installations at the same time.
- The **nim_master_setup** and **nim_clients_setup** scripts provide a way to set up the NIM environment.
- Requires an available server running AIX that can be used as the NIM master.

1. Run the **oslevel** command on the Master_LPAR. Output similar to the following displays:

   ```
   5200
   ```

   If the output from the **oslevel** command does not show the expected OS level, see "Migrating AIX" on page 405 for information about migrating the AIX operating system to the correct OS level.
2. Verify your network connection by running the **netstat** command. You can run this command with the **-C** flag to show routing table information. You can also use the **-D** flag to show the number of packets received, transmitted, and dropped in the communications subsystem.
3. Insert the AIX installation media.
4. Run the **nim_master_setup** command. For additional information on options for running this command, see **nim_master_setup**.
5. Run the following command to open the`/export/nim/client.defs` file with the vi editor:

   ```
   # vi /export/nim/client.defs
   ```
6. Edit the `client.defs` file according to your environment. For more information on this file, see the instructions and examples in the `client.defs` file. When you are finished editing the `client.defs` file, save it and exit the vi editor.
7. Run the **nim_clients_setup -c** command. For additional information on options for running this command, see **nim_clients_setup**.

   Note: If you are adding new client machines that cannot be resolved on the name server, edit the `/etc/hosts` file to add the IP addresses and client host names.

After you have defined the client machines, you should activate and install the partitions.

After you activate and install the client partitions, you can perform any system management tasks.

Related information:

Activating a partition profile

Updating your NIM environment to the latest technology level

The **nim_update_all** command provides a one-step method to update an existing NIM environment and any resources created with the **nim_master_setup** command to the latest technology level.

1. Insert the latest AIX update media into the media drive.
2. Update the bos.rte.install fileset by running the **geninstall** command as follows:
   ```
   # geninstall -d /dev/cd0 bos.rte.install
   ```
3. Run the **install_all_updates** command as follows:
   ```
   # install_all_updates -d /dev/cd0
   ```

   The output from the **install_all_updates** command is shown in the /var/adm/ras/install_all_updates.log log file. The **install_all_updates** command checks whether your system is at the latest known technology level. If your system is not at the latest known technology level, your server is updated to the latest technology level of AIX.
4. After the update is complete, reboot the system by using the **shutdown -Fr** command.
5. Run the **nim_update_all** command to update any NIM resources created by the **nim_master_setup** command. The **nim_update_all** command uses the device **/dev/cd0** by default. The output from the **nim_update_all** command is shown in the /var/adm/ras/nim.update log file. Output similar to the following displays:
   ```
   ############################ NIM update all ############################
   #                                                                      #
   #  During script execution, NIM client and resource updating times     #
   #  may vary. To view the install log at any time during nim_update_all,#
   #  run the command: tail -f /var/adm/ras/nim.update in a separate screen.#
   #                                                                      #
   ########################################################################

   NSORDER=local,bind
   Adding updates to lpp_res lpp_source....done
   Updating spot_res using updated lpp_source lpp_res....done

   Attempting to replace mksysb resource generic_sysb...
   Removing old mksysb resource generic_sysb....done
   Creating image.data file....done
   Checking /export/nim space requirement...

   Generating list of client objects in NIM environment...
   ```

A new mksysb is created that replaces the existing mksysb, unless you specify the **-B** flag. All clients in the environment are updated, unless you specify the **-u** flag.

Advanced installation options

Advanced installation options are described in the /usr/lpp/bos/README.PARTITION_INSTALL file.

Some examples of advanced installation options are the following:

- Preparing an existing root volume group (**rootvg**) to move to a managed system
- Using the **devreset** command to rebuild the device ODM database and reset all devices to default configurations

Creating and changing a dedicate dump device

You can create a dedicated device where system crash data is dumped.

To create and change to a dedicated dump device, do the following:

1. Determine the size of the hd6 paging space (in logical partitions) by running the **lsvg** command as follows:
   ```
   # lsvg -l rootvg
   ```
   The output will be similar to the following:
   ```
   rootvg:
   LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
   hd5                 boot       1     1     1    closed/syncd  N/A
   ```

```
hd6                 paging     8     8     1    open/syncd    N/A
hd8                 jfs2log    1     1     1    open/syncd    N/A
hd4                 jfs2       1     1     1    open/syncd    /
hd2                 jfs2       12    12    1    open/syncd    /usr
hd9var              jfs2       1     1     1    open/syncd    /var
hd3                 jfs2       1     1     1    open/syncd    /tmp
hd1                 jfs2       1     1     1    open/syncd    /home
hd10opt             jfs2       2     2     1    open/syncd    /opt
hd11admin           jfs2       2     2     1    open/syncd    /admin
livedump            jfs2       4     4     1    open/syncd    /var/adm/ras/livedump
```

In this example, the paging space is 12 LPs (logical partitions) in size.

2. Create a dump logical volume by running the **smitty mklv**. When you are prompted for the volume group, type `rootvg`.
3. In the **Add a Logical Volume** menu, fill in the **Logical volume NAME** and the **Number of LOGICAL PARTITIONS** fields.
4. Change the primary dump device by running the **smitty dumpchgp**. You are prompted to enter the path to the primary dump device.
5. Validate your dump devices by running the **smitty dump** command.
6. Select **Show Current Dump Devices**. The output will be similar to the following:

```
                                 COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

primary              /dev/dumplv
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
```

Verifying your dump device

If your machine has more than 4 GB of real memory, a dedicated dump device is created at installation time. Otherwise, the `/dev/hd6` paging space is used as the dump device.

If a system crash occurs and paging space was used as the dump device, the dump is copied to the `/var/adm/ras/vmcore.`*n* file by default, where *n* is a sequence number. If there is not enough space to perform the copy, the user is prompted during reboot to save the dump to some other media. To avoid losing a dump due to a lack of a tape drive configured to the partition, always create a separate dump device that is the same size as your paging space, given that paging space is currently your dump device.

To verify your dump device, type `smitty dump`, and select **Show Current Dump Devices**. If paging space is your dump device, the output will be similar to the following:

```
                                 COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

primary              /dev/hd6
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
```

Shutting down a partition

You can shut down an AIX partition to return it to the *Ready* state.

When a partition is up, it is in the *Running* state. To use AIX to shut down a partition, complete the following steps:

1. Type `shutdown -Fr` on the AIX command line. The partition changes to the *Starting* state, and the operator panel values display, which indicates AIX is now rebooting. When the reboot is issued within AIX, the partition will boot according to the contents of the bootlist. To check the order of the boot devices, type `bootlist -m normal -o`.
2. Type `shutdown -F` to shut down AIX.

The partition will eventually change to the *Ready* state. You have now shut down AIX and its partition.

Changing your operating system host name

Each partition, including the Full System Partition, must have a unique host name that can be resolved. If you want to change the host name of a partition, you must also change the operating system host name.

Host names cannot be reused between the Full System Partition and the logical partitions. To change the operating system host name, complete the following steps:

1. Run the **lsrsrc** command as follows:

   ```
   /opt/rsct/bin/lsrsrc ManagementServer Hostname
   ```

   If the partition is managed by multiple HMCs, multiple entries might exist because each HMC has its own entry. The output will be similar to the following:

   ```
   resource 1:
            Hostname       = "hmc1.mydomain.mycompany.com"
   ```

2. For each entry, use the **rmrsrc** to remove the host name shown. For example, run the following command:

   ```
   /opt/rsct/bin/rmrsrc -s 'Hostname = "hmc1.mydomain.mycompany.com"' ManagementServer
   ```

   You can verify that all entries have been removed by running the **lsrsrc** command again.
3. Run the **rmcctrl** command as follows:

   ```
   /opt/rsct/bin/rmcctrl -z
   ```

4. Change the host name of the partition.
5. Run the following command:

   ```
   /opt/rsct/bin/rmcctrl -A
   ```

Notices

This information was developed for products and services offered in the US.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide 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:

© Copyright IBM Corp. 2015, 2018

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Index

Special characters

/dev directory 331
/dev/ipldevice file 95
/etc/exclude.rootvg file 321
/etc/niminfo file
  recovering 195
/etc/objrepos directory
  after installing from system backup 329
/file system
  root part 401
/tmp directory or /tmp subdirectories
  creating NIM resources 230
/tmp file system
  free space in 21, 319
  messages 95
  size during installation from backup 329
/tmp/disk.image file 95
/tmp/unix file 95
/tmp/vgdata/rootvg directory 331
/usr file system 253
  messages 95
  troubleshooting when full 94
  user part 401
/usr/share file system 401

A

accessing an unbootable system 91
accessing SMIT 339
activate operation 257
Activation Engine Overview 33
adapter_def resource 229
  defining 229
adding BCMM management
  NIM environment 132
adding CEC management
  NIM environment 129
adding HMC management
  NIM environment 128
adding IVM management
  NIM environment 131
adding nas_filer management
  NIM environment 133
adding PowerVC management
  NIM environment 134
adding VIOS management
  NIM environment 130
additional topics
  dynamic host configuration protocol
    interacting with 121
advanced configuration 214
  adding another network type 189
  backing up the NIM database 195
  booting diagnostics 192
  booting in maintenance mode 191
  creating additional interface attributes 161
  defining /usr vs. non-/usr SPOTs 150
  defining a heterogeneous network 187
  defining an lpp_source on CD/DVD-ROM vs. Hard Disk 152
advanced configuration *(continued)*
  establishing a default route 189
  establishing a static route 190
  recovering the /etc/niminfo file 195
  removing machines 196
  restoring the database and activating the master 196
  unconfiguring the master 150
advanced installation
  controlling the master or client 120
  group member
    excluding 212
    including 212
  logs
    viewing boot 208
    viewing configuration 208
    viewing installation 208
  lpp_source
    copying software 220
    maintaining software 220
    removing software 221
    running the check operation 221
  lppchk operation
    verifying installation 183
  machine groups
    adding new members 212
    defining 211
    removing members 197
  machines
    resetting 210
  resource group
    allocating 231
    defining 231
  resource groups
    defining default 232
  resource servers
    using clients 163
  SPOT
    listing software 218
    listing software updates by keyword 218
    maint operation 219
    managing software 217
    rebuilding network boot images 210
  standalone client
    listing software 218
    listing software updates by keyword 218
  standalone clients
    maint operation 219
    managing software 217
advanced tasks
  installation 211
affinity logical partitions
  overview 421
AIX Relocatable Installation 62
AIX service enhancement 338
all devices and kernels 50
allocate
  resource group 231
allocate operation 258
allocating resources
  resource groups 231
alt_disk_install 174

© Copyright IBM Corp. 2015, 2018

alternate disk installation 68
cloning 69
data access 74
dynamic logical partitioning 74
examples 75
in a partitioned environment 432
mksysb installation 68
phased installation 69
using SMIT 74
Alternate Disk Installation 13
Alternate Disk Migration Installation 70
American Standard Code for Information Interchange 85
applying optional software
definition of 343
description 335
ASCII Installation Assistant 85
introduction to tasks 85
ASCII procedures 335
ASCII terminals
setting communications options 331
setting display and keyboard options 58, 331
setting options 58
ATM networks
installing to clients 139
ATM Networks 185
converting generic networks into 140
attributes
if 161
if1 161
if2 161
other_net_type 187

B

backup
disk space 329
backup image 329
backup installation
changing installation modes 331
backup, of system 318
exclude files 321
introduction 322
mounting and unmounting file systems 21, 319
procedure
list information about filesets in a system image 329
lsmksysb command 328
lssavevg command 328
prerequisites 21, 319
preview information about a backup 328
root volume group 21, 320
verifying backup 327
verifying system backups 329
view the backup log 329
basic configuration
creating basic installation resources 121
basic host identification 215
basic operations and configuration 121, 174
adding a diskless or dataless client 147
adding a stand-alone client 128
adding a standalone client 134
clients and SPOT resources
customizing 141
configuring the master 121
diskless and dataless clients
configuring the master and creating resources 144
diskless or dataless machine
initializing and booting 193
basic operations and configuration *(continued)*
diskless or dataless machine *(continued)*
uninitializing 148
mksysb install 169
non-prompted install 168
rte install 167
binary compatibility 408
boot logs
view 208
boot resource 233
booting
diagnostics 192
over router 195
booting the system
problems with 91
procedure 57, 331, 412
BOS installation
cloning 22, 331
system backup, from 329
BOS maintenance mode
accessing 91
BOS menus
installation and setting window 41
installation summary window 41
welcome window 41
BOS run-time image
source for 265
bos_inst operation 262
ATM adapters 139
paging space 184
using the boot_client attribute 266
using the force_push attribute 266
using the preserve_res attribute 267
using the set_bootlist attribute 267
bos.sysmgt.nim.master
removing 221
bosboot
troubleshooting 95
device problems 95
space problems 95
bosinst_data 233
defined 233
defining 234
overview 233
bosinst.data control_flow stanza descriptions 46
bosinst.data file 46
ACCEPT_LICENSES variable 49
ALL_DEVICES_KERNELS variable 50
ALT_DISK_INSTALL_BUNDLE variable 50
ALWAYS_ALLOW variable 54
BOSINST_DEBUG variable 49
BOSINST_LANG variable 54
bosinst.data target_iscsi_data stanza 53
BUNDLES variable 48
CONNECTION variable 52
CONSOLE variable 46
COPYDIR variable 54
CULTURAL_CONVENTION variable 54
CUSTOMIZATION_FILE variable 48
description 81
DESKTOP variable 49
DUMPDEVICE variable 54
ERASE_ITERATIONS variable 50
ERASE_PATTERNS variable 50
ERROR_EXIT variable 48
EXISTING_SYSTEM_OVERWRITE variable 47
FORCECOPY variable 54

bosinst.data file *(continued)*
GRAPHICS_BUNDLE variable 50
HARDWARE_DUMP variable 50
HDISKNAME variable 51
IMPORT_USER_VGS variable 50
INSTALL_DEVICES_AND_UPDATES variable 50
INSTALL_EDITION variable 47
INSTALL_METHOD variable 47
INSTALL_TYPE variable 48
INSTALL_X_IF_ADAPTER variable 48
KERBEROS_5S_BUNDLE variable 50
KEYBOARD variable 54
livedump stanza 55
LOCATION variable 51
MESSAGES variable 54
mksysb_migration_device 57
nonprompted BOS installation 47
PHYSICAL_LOCATION variable 51
PRIMARY variable 54
PROMPT variable 47
prompted mode, with 92
PVID variable 51
RECOVER_DEVICES variable 49
REMOVE_JAVA_5 variable 50
RM_INST_ROOTS variable 48
RUN_STARTUP variable 48
SAN_DISKID variable 52
SECONDARY variable 54
SECURE_BY_DEFAULT variable 49
SERVER_BUNDLE variable 50
SIZE_MB variable 51
SIZEGB variable 54
Specifies the edition selection 47
SYSTEM_MGMT_CLIENT_BUNDLE variable 49
TRUSTED_AIX variable 49
TRUSTED_AIX_LSPP variable 49
using 82
bosinst.data sample file 56
build date of filesets installed 338
bundles 335, 403
definition of 335
examples of 403
types of 403

C

CD or DVD
creating backups CD or DVD 324
CD/DVD
bootable 322
non-bootable 322
cdrecord 322
change characteristics 118
change operation 267
check operation 267
checking 118
chwpar operation 268
cleaning up failed software installation 23, 345
client
determining control 120
diskless or dataless
adding 147
operations 108
tasks performed from 214
client communication options management
NFS
defining 214
client machine
verifying status of 138
client operations 214
clients
dataless 113
initializing 114
diskless 113
initializing 114
optional resources 113
required resources 113
shared_root 252
standalone 112
managing software 113
network booting 113
standalone, adding 134
cloning
backup, of system 22, 331
cloning using Alternate Disk Installation 13
commands
nim_clients_setup 119
nim_master_setup 119
nimclient 214
committing service updates
introduction 335
Common Criteria 44
communication between HMC and Partitions 423
communications
ASCII terminals options 58
setting options for ASCII terminals 331
compatibility 408
concepts
NIM 107
concurrency control 164
configuration assistant
web browser, installing 85
web server, installing 85
Configuration Assistant
configuring online documentation 85
documentation 85
configuration logs
view 208
configuration menus
using iSCSI 66
configuration tasks
basic 121
configuration, system 85
access remote resources 85
add license passwords 85
back up the system 85
change language environment 85
configure printer 85
create user accounts 85
date and time 85
exit and log in 85
install optional software 85
introduction to 85
list of tasks 85
set root password 85
consolidation of multiple versions of the same operating system 423
control operations
definition of 257
control status
master or client 120
control_flow stanza 46
Create and Install a Software Bundle
software bundle 9, 16

create backup 118
Creating AE scripts 38
Creating AE Template File 35
creating clients 117
Creating EZNIM setup using different 283
creating software packages 402
cust operation 268
  asynchronous behavior 222
customizing BOS installation 82
  bosinst.data file 81

D

data recovery
  introduction 91
  procedure 92
  when system will not boot 91
dataless clients 113
  initializing 114
  managing software 115
  shared_root 252
deactivate operation 270
deallocate operation 270
debug
  BOS installation 49
debug mode
  installing BOS 41
  NIM 315
  NIM BOS installation 316
  using a bosinst.data file for NIM BOS installation 317
default routes
  creating 189
define
  machine group 211
  resource group 231
define operation 270
definition file
  KDC server sample 287
  Kerberos slim client sample 287
devexports resource 234
  defining 234
  overview 234
diag operation 192, 271
diagnostics
  booting 192
  loading from network
    for diskless and dataless machines 193
directories
  /dev 331
  /etc/objrepos 329
  /tmp/vgdata/rootvg 331
disk
  specifying for BOS installation
    CD-ROM, DVD-ROM or tape 59
    system backup 331
disk space
  messages 95
  planning for master machine 121
diskless and dataless
  managing software 115
diskless and dataless clients
  creating resources to support 144
  migrating 211
diskless and dataless tasks
  booting diagnostics 192
diskless clients 113
  initializing 114
diskless clients *(continued)*
  managing software 115
  resource
    boot 113
    dump 113
    home 113
    paging 113
    resolv_conf 113
    root 113
    shared_home 113
    SPOT 113
    tmp 113
  resources
    optional 113
    required 113
diskless or dataless client
  adding 147
  initializing and booting 193
  uninitializing 148
disks, hard (fixed) 59
displays
  setting options for ASCII terminals 58, 331
distributed resources 230
dkls_init operation 271
DLPAR
  Live Update operation 391
dtls_init operation 272
dump device
  verifying 436
dump devices
  creating and changing 435
  in a partitioned environment 435
dump resource 235
  defined 235
  defining 235
  overview 235
DVD-RAM
  creating backups DVD-RAM and UDF 325
dynamic host configuration protocol
  interacting with 121
dynamic logical partitioning 421

E

electronic license agreements 43
environment
  changing language 59
epkg command 368
error conditions 87, 288
error messages
  attempt to create bootable tape failed 95
  check available disk space 95
  error occurred during bosboot 95
  format 95
  hard disks not accessed 95
  hard disks not configured 95
  invalid or no boot device specified 95
  NIM 288
  no disks are available 95
  not enough file space to create: /tmp/disk.image 95
  not enough file space to create: /tmp/unix 95
  unable to expand file system /usr 95
error recovery 87, 288
Evaluation Assurance Level 4+ 44
exclude files from system backup 321
exclude_files 236
  defined 236

exclude_files *(continued)*
defining 236
overview 236
exporting resources
number of hosts 222
eznim 282

F

fb_script 236
defined 236
defining 237
overview 236
FDDI
router 195
file system
messages 95
file systems
/
messages 95
root part 401
/tmp 329
free space in 21, 319
messages 95
/usr
messages 95
troubleshooting when full 94
usr part 401
/usr/share 401
mounting and unmounting 21, 319
file_res resource 225
files
/dev/ipldevice 95
/etc/exclude.rootvg 321
/etc/niminfo 195
/tmp/disk.image 95
/tmp/unix 95
/usr 253
bosinst.data 82
examples of 56
explanation of 81
prompted mode, with 92
examples
bosinst.data 56
definition file for nimdef command 285
script resource 284
image.data 82, 329
map 331
preserve.list 82
sample 284
filesets 335, 401
fix_bundle 237
defined 237
defining 237
overview 237
fix_query operation 273
fixed disks 59
fixes
listing for SPOT 218
listing for standalone client 218
full system partition 421

G

geninstall command 350, 400
graphical user interface
Easy Install
prerequisites 338
procedural overview 335
graphical user interfaces
Installation Assistant
introduction to tasks 85
group
resource
allocate 231
define 231
groups
establishing 226
machine 226
defining 211
resource 227
GUIs 85

H

hard disks 59
location codes of 59
specifying for CD/DVD-ROM or tape installation 59
specifying for system backup installation 331
unaccessible 95
unconfigured 95
hardware
diagnostics
diskless and dataless 192
Hardware Management Console (HMC)
affinity logical partitions 421
communicating with partitions 423
managing remotely with client 422
managing systems with 420, 422
managing with an HMC client 422
Network Adapter Communication 423
heterogeneous networks
defining 187
home resource 238
defined 238
defining 238
overview 238
host authentication 215
host authorization 215
host identification 215
host name
naming 437

I

if attribute
creating additional 161
if1 attribute 161
if2 attribute 161
image_data 239
defined 239
defining 239
overview 239
image.data file 82, 329
install_all
SMIT fast path 339
install_all_updates command 342
installation
alternate disk 68
change method of 59
verifying with lppchk operation 183

Installation Assistant 86
introduction to tasks 85
installation images
mksysb configuration on backup tape 88
installation logs
view 208
installation methods
definition of 59
specifying 59
installation resources
creating 121
installation screens (BOS)
Change Disk(s) Where You Want to Install 331
specifying installation language 413
specifying system console 331, 413
System Backup Installation and Settings (BOS) 331
installation tasks
advanced 211
installation windows (BOS)
Change Disk(s) Where You Want to Install 59
Installing Base Operating System 61, 414
specifying installation language 59
specifying system console 59
installation, nonprompted
specifying with the bosinst.data file 81
installing AIX
considerations 24, 28, 424, 428
in a partitioned environment 24, 28, 424, 428, 432, 434
NIM 432, 434
procedures 24, 28, 424, 428
using the media to manually install a logical partition 24, 28, 424, 428
using the media to manually install a standalone system
new and complete overwrite installation 2
installing and managing interim fixes 354
installing and managing software
detached WPAR 224
installing BOS
debug mode 41
new and complete overwrite 57
preservation 57
installing BOS from CD or DVD-ROM
procedure
specifying destination disk 59
installing BOS from CD or tape
procedure
booting (starting) the system 331
initiating the installation 331
installing BOS from CD-ROM
procedure
changing language environment 59
prerequisites for 57
installing BOS from CD/DVD-ROM
procedure
booting (starting) the system 57
initiating the installation 57
introduction to 57
installing BOS from CD/DVD-ROM or tape
procedure
booting (starting) the system 412
initiating the installation 412
introduction to 411, 412
installing BOS from DVD-ROM
procedure
prerequisites for 57
installing BOS from media or tape
troubleshooting 95
installing BOS from system backup
introduction to 329
procedure 331
resolving reported problems 89
source system 329
target system 329
troubleshooting 87
installing optional software
applying 335, 343
cleaning up failed installation 23, 345
committing 335
introduction to 335
prerequisites 338
procedural overview 335
procedure (SMIT) 339
status messages 341
rejecting 335, 344
removing 335, 344
selection criteria 337
software licenses 337
software packaging 337
status messages (SMIT) 341
troubleshooting 23, 95, 345
installing software 118
Installing to an alternate disk 13
installp format
creating software packages 402
installp_bundle 239
defined 239
defining 239
overview 239
InstallShield MultiPlatform 350
installing a package 350
silent installation 351, 354
uninstalling a package 351
interface attribute 161
creating additional 161
interim fix 354
Live Update 375
interim fix management
emgr command 354
interim fix control file 368
see also interim fix management 354
interim fixmanagement
epkg command 368
introduction to NIM
network objects
definition of 107
IP address
determining 186
iSCSI configuration menus 66
ISMP
see InstallShield MultiPlatform 350
ISO9660 format 324

K

KDC server
sample definition file 287
Kerberos
host identification 215
NFS V4 host authentication 215
Kerberos slim client
sample definition file 287
kernel extension
loading 398

keyboards
setting options for ASCII terminals 58, 331

L

language environment
changing 59
setting 59
license acceptance 49
license agreements 43
licensed programs 335
packaging of 337
selection criteria for installation 337
licenses, software
function of 337
list characteristics 118
listing fixes
installed on a SPOT 218
installed on a standalone client 218
listing information
software in SPOT 218
software in standalone client 218
listing software updates
installed on a SPOT 218
installed on a standalone client 218
Live Update 376
configurations 384
customization 391
defining live_update_data_resource 245
interim fix installation 375
kernel extension 398
limitations 378
notifications 391
overview 376
preparing 378
prerequisites 388
preview mode 389
process classification 396
safe kernel extension 398
system tunables 396
timeline for scripts execution 393
using live_update_data_resource 244
via geninstall command 390
via NIM 389
locale 59
location codes, of the hard disk 59
logical partition
consolidation of multiple versions of the same operating system 423
implementation 422
minimum resources 420
mixed production and test environments 423
overview 420
running AIX 421
server consolidation 423
logical volumes
accessing 91
introduction 91
procedure 92
logs
boot
viewing 208
configuration
viewing 208
installation
view 208
lpp_source 241
lpp_source *(continued)*
copying software 220
defined 241
defining 243
maintaining software 220
overview 241
removing software 221
running the check operation 221
lppchk operation 273
lppmgr command 345
lppmgr operation 274
lsmksysb command 328
lssavevg command 328
lswpar operation 274

M

machine
diskless or dataless
initializing and booting 193
uninitializing 148
machine groups
defining 211, 226
operations 212
machine operations
defined 257
list of 257
machines 108
maint operation 219, 274
asynchronous behavior 222
maintain software in SPOT 219
maintain software on standalone clients 219
maint_boot operation 275
maintaining optional software
applying 343
concepts defined 343, 346
rejecting 344
removing 344
maintenance mode
accessing (BOS) 91
maintenance mode recovery 67
maintenance mode to recover iSCSI parameters 67
maintenance updates
automated downloads 346
management 354
managing clients 116
Managing NFS client communication options 214
Managing NFS client communication options using SMIT 217
Managing NFS client communication options using the command line 217
managing software
clients
standalone 113
SPOT 217
standalone clients 217
map files 331
master
activating 196
backing up NIM database 195
configuring 121
disabling push permissions 209
managing
activate the master 196
back up NIM database 195
restore NIM database 196
removing master fileset 221
unconfiguring 150

master fileset
removing 221
messages
NIM error 288
NIM warning 288
system and error 95
migrating
clients
diskless and dataless 211
multibos instance 414
NIM SPOTs 211
migrating installing AIX 6
migrating to new version
procedure
prerequisites for 411
migration installation
definition of 39
mkcd command 323
mkinstallp command 402
mkisofs 322
mksysb 245
alternate disk installation 174
backup images on CD/DVD 322
cloning 22, 331
defined 245
defining 245
installation from 329
overview 245
resolving reported problems 89
source for BOS run-time image 265
system backup tapes 88
troubleshooting installation from 87
mksysb install
performing 169
mksysb_migration_device 57
monitors
setting options for ASCII terminals 58, 331
multibos instance 414
multibos utility 76

N

name resolution 120
naming your OS host 437
network
heterogeneous
defining 187
Network Adapter Communication Between Partitions and the HMC 423
network booting
clients
standalone 113
network objects
managing
creating interface attributes 161
establishing a route between networks 190
network types
supported 185
networks
ATM 139
defining 185
defining heterogeneous 187
NIM 185
new and complete overwrite installation 57
definition of 39
NFS
client communication options management 214
NFS client communication options
managing using SMIT 217
managing using the command line 217
NFS V4 host authentication 215
NFS V4 host authorization 215
NFS V4 host identification 215
NIM
adding WPAR clients 138
alternate disk installation 174
client requests, tuning 149
configuration
basic 121
configure using EZNIM 15
dataless
definition of 108
diskless
definition of 108
error messages 288
exported Kerberos authentication 179
in a partitioned environment 432, 434
Kerberos 5 179
Kerberos authentication 178
machines 108
networks 185
nimesis daemon 149
operations
basic 121
overview 107
resources
definition of 225
response files and InstallShield MultiPlatform products 354
SPOTs
migrating 211
standalone
definition of 108
warning messages 288
NIM attributes 187
NIM clients
defining 108
NIM concepts 107
NIM database
backing up 195
restoring 196
NIM environment 108, 116
defining
using the nimdef command 213
NIM eznim 282
NIM groups 226
NIM networks
defining 185
IP address
determining 186
routes 187
types
supported 185
NIM object definitions
name requirements 120
NIM objects
definitions
name requirements 120
NIM operations 192
activate 257
allocate 258
alternate disk installation 228
alternate disk migration installation 228
bos_inst 262

NIM operations *(continued)*
change 267
check 267
chwpar 268
cust 268
deactivate 270
deallocate 270
define 270
diag 271
diagnostics, booting 228
diskless and dataless clients, adding 228
dkls_init 271
dtls_init 272
fix_query 273
lppchk 273
lppmgr 274
lswpar 274
maint 274
maint_boot 275
maintenance mode, booting 228
reboot 276
remove 276
reset 276
resources 228
select 277
showlog 277
showres 278
software, customizing 228
software, removing 228
sync 279
sync_roots 279
syncwpar 279
takeover 280
unconfig 280
update 281
updateios 281
NIM output
suppressing 222
NIM resources 225
NIM routes 187
NIM Service Handler 214
nim_script resource 246
nim_update_all
in a partitioned environment 435
nimclient command 214
nimdef command
NIM environment
defining 213
sample definition file 285
NIMSH 214
non-prompted install
performing a 168
nonprompted installation
changing to prompted 331
nonprompted mode
overriding 92
nonroot volume group
definition of 318

O

operations
activate 257
allocate 258
basic 121
BOS run-time image
selecting source 265
operations *(continued)*
bos_inst 262
change 267
check 267
chwpar 268
cust 268
deactivate 270
deallocate 270
define 270
diag 271
dkls_init 271
dtls_init 272
excluding a group member 212
fix_query 273
including a group member 212
list of 257
lppchk 273
lppmgr 274
lswpar 274
maint 274
maint_boot 275
NIM
diag 192
machine 257
performed from client 214
on client machines 108
performing 257
reboot 276
remove 276
reset 276
select 277
showlog 277
showres 278
sync 279
sync_roots 279
syncwpar 279
takeover 280
types 257
unconfig 280
update 281
optional software
cleaning up failed installation of
introduction 23, 345
definition of 335
options
language environment 59
nonprompted installation
specifying with the bosinst.data file 81
setting communications (ASCII) 58, 331
setting display (ASCII) 58, 331
setting monitor (ASCII) 58, 331
specifying installation disk (BOS)
CD-ROM, DVD-ROM or tape 59
system backup 331
specifying installation language 59, 413
specifying installation method (BOS) 59
specifying system console 59, 331, 413
system configuration 85
other_net_type attribute 187
output, NIM
suppressing 222
overview
NIM 107
overview of partitions
managed system 420

P

package
  definition of 335
package formats
  installp 400
    software filesets 401
packaging interim fixes 368
paging
  defined 247
  overview 247
paging resource 247
  defining 247
partition
  security 422
partition scenarios
  Advanced Installation Options 435
partitions
  affinity 421
  full system 421
  implementation 422
  logical 420
  running AIX 421
  using NIM 432, 434
port conflicts
  nimesis daemon 317
post_migration command 410
pre_migration command 410
prerequisites for setting up a NIM environment with NFS security using Kerberos 5 216
preservation installation 57
  definition of 39
preserve_res 267
preserve.list file 82
problems, recovering from 87, 288
procedures
  accessing BOS maintenance 92
  advanced configuration 214
    adding another network type 189
    backup up the NIM database 195
    booting diagnostics 192
    booting in maintenance mode 191
    creating additional interface attributes 161
    defining a heterogeneous network 187
    defining an lpp_source on CD/DVD-ROM vs. Hard Disk 152
    establishing a default route 189
    establishing a static route 190
    recovering the /etc/niminfo file 195
    removing machines from the NIM environment 196
    restoring the database and activating the master 196
    unconfiguring the master 150
  advanced installation
    adding new members 212
    allocating a resource group 231
    controlling the master or client 120
    copying software to an lpp_source 220
    defining a machine group 211
    defining a resource group 231
    defining default resource groups 232
    excluding a group member 212
    including a group member 212
    listing software in a SPOT 218
    listing software on a standalone client 218
    listing software updates by keyword 218
    maintaining software in an lpp_source 220
    maintaining software on a SPOT 219
    removing members 197
procedures *(continued)*
  advanced installation *(continued)*
    removing software from an lpp_source 221
    resetting machines 210
    running the NIM check operation 221
    standalone clients and SPOT resources, managing software 217
    using clients as resource servers 163
    verifying installation with lppchk operation 183
    viewing logs 208
  advanced installation, managing software
    rebuilding network boot images for a SPOT 210
  basic configuration
    creating basic installation resources 121
  basic operations and configuration
    adding a diskless or dataless client 147
    adding a standalone client 134
    clients and SPOT resources, customizing 141
    configuring the master 121
    creating resources to support diskless/dataless clients 144
    diskless or dataless machine, initializing and booting 193
    diskless or dataless machine, uninitializing 148
    mksysb install, performing 169
    non-prompted install 168
    rte install, performing an 167
  defining /usr vs. non-/usr SPOTs 150
  identifying boot device 95
  installing BOS from CD/DVD-ROM 57
  installing BOS from CD/DVD-ROM or tape 411, 412
  installing BOS from system backup 331
  installing optional software 335
  resizing /tmp 95
  root volume group, backing up 21, 320
  troubleshooting a mksysb installation 87, 89
  troubleshooting full /usr file system 94
  unlocking the root volume group 95
  user volume group, backing up 327
  verifying system backup 329
product identification, optional software 336
prompted installation 334
  help information 334
prompted mode
  changing to (BOS) 92
push permissions
  master
    disabling 209

R

readme
  README.PARTITION_INSTALL 435
reboot operation 276
recover devices 49
recovery 87, 288
  maintenance mode 67
rejecting optional software
  definition of 344
  introduction 335
relocatable application execution 65
relocatable applications packaging 64
Relocatable Installation 62
relocatable installation utilities on AIX 63
remote management 422
remove operation 276

removing optional software
definition of 344
introduction 335
reset operation 276
resolv_conf
defined 247
defining 247
overview 247
resolv_conf resource 247
resource
boot
diskless/dataless 113
dump
diskless/dataless 113
home
diskless/dataless 113
paging
diskless/dataless 113
resolv_conf
diskless/dataless 113
root
diskless/dataless 113
shared_home
diskless/dataless 113
SPOT
diskless/dataless 113
tmp
diskless/dataless 113
resource group
allocate 231
define 231
resource groups
allocating resources 231
defining 227
defining default 232
resources 225
adapter_def 229
defining 229
AIX Version 4.3 or later spots 254
boot 233
bosinst_data 233
defining 234
clients and SPOT
customizing 141
devexports 234
defining 234
distributed 230
dump 235
defining 235
exclude_files 236
defining 236
exporting
number of hosts 222
fb_script 236
defining 237
file_res 225
fix_bundle 237
defining 237
home 238
defining 238
image_data 239
defining 239
installp_bundle 239
defining 239
list of 225
lpp_source 241
defining 243
resources *(continued)*
mksysb 245
defining 245
nim_script 246
operations
list of 225
paging 247
defining 247
resolv_conf 247
defining 247
root 248
defining 248
savewpar 249
defining 249
script 249
defining 249
secattrs 250
defining 250
shared product object tree 253
shared_home 251
defining 251
SPOT 253
defining 254
tmp 257
defining 257
wpar_spec 255
defining 255
resources, NIM
reducing space requirements 223
response files 351
response files and NIM 354
root
shared_root 252
root directory
creating file resources 230
root resource 248
defined 248
defining 248
root volume group (rootvg)
backing up 21, 320
definition of 318
unlocking 95
router
booting over 195
routes 187
creating a default 189
creating a static 190
RPM
ISMP 400
rte
source for BOS run-time image 265
rte install
performing 167
run-time image 265

S

sample files 284
bosinst.data 56
nimdef command
definition file 285
script resource 284
sample KDC server definition file 287
sample slim client definition file 287
savevg command 327
savewpar resource 249
defining 249

savewpar resource *(continued)*
overview 249
screens
Access a Root Volume Group (BOS) 92
Change Disk(s) Where You Want to Install (BOS) 331
Maintenance (BOS) 92
System Backup Installation and Settings 331
Volume Group Information (BOS) 92
Welcome to Base Operating System Installation and Maintenance (BOS) 92
script 249
defined 249
defining 249
overview 249
script resource
sample file 284
secattrs resource 250
defining 250
overview 250
security
Controlled Access Protection Profile and Evaluation Assurance Level 4+ 422
security evaluation technology 44
select operation 277
service update management assistant 346
service updates
rejecting 344
removing 344
setting up a NIM environment with NFS security using Kerberos 5 prerequisites 216
shared product object tree 253
shared volume groups
AUTO ON 329
shared_home
defining 251
overview 251
shared_home resource 251
defined 251
showlog operation 277
showres operation 278
shutting down a partition
using AIX 437
SMIT fast path
alt_clone 74
alt_mksysb 74
assist 86
cleanup_software 349
compare_report 347
eznim 282
fixtolist_compare 348
install_all 339
install_latest 339
install_update 86
installed_license 337
instofix_compare 348
instolist_compare 348
license_on_media 337
reject 344
remove 344
rename_software 349
service_software 346
update_all 339
update_by_fix 339
SMIT interfaces
Custom Install path 339
Easy Install path 339
prerequisites 338
SMIT interfaces *(continued)*
Easy Install path *(continued)*
procedural overview 335
Installation Assistant 85
SMIT procedures
installing optional software 335, 339
software
listing for SPOT 218
listing for standalone client 218
software bundles 335, 403
definition of 335, 403
examples of 403
software filesets
definition of 335
software licenses
function of 337
software packages
definition of 335
software packaging
root part 401
share part 401
user part 401
software products 337
applying 343
bundle, definition of 335
bundle, examples of 403
fileset, definition of 335
identification of 336
licensed program, definition of 337
package, definition of 335
packaging of 337
rejecting 344
removing 344
selection criteria for installation 337
software service management 346
clean up software images 349
comparison reports 347
rename software images 349
software updates
listing for SPOT 218
listing for standalone client 218
software, optional
definition of 335
source system 329
SPOT
defined 253
defining 254
maintaining software 219
managing software 217
overview 253
resources 151
source for BOS run-time image 265
updating 214
standalone client operations 214
standalone clients 112, 214
adding 134
maintaining software 219
managing software 113, 217
network booting 113
standalone machines
determining control 120
standby BOS 78
stanza
bosinst.data target_iscsi_data 53
starting 117
starting the system 57, 331, 412

static routes
creating 190
stopping 118
suppressing
NIM output 222
sync operation 279
sync_roots operation 279
synchronizing software 118
syncwpar operation 279
system
booting (starting) 57, 331, 412
configuring 85
system backup 318
system backup to tape 21
system backup, BOS installation from 329
system bundles 403
system configuration 85
System Management Interface Tool 85
system messages 95
system settings
changing during BOS installation 59
system, managed 420

T

takeover operation 280
tape
creating bootable, troubleshooting 95
creating bootable,troubleshooting 95
tapeblksz 88
target disk
specifying for CD/DVD-ROM or tape installation 59
specifying for system backup installation 331
target system 329
target_iscsi_data stanza 53
tasks
bos, installing 228
configuring NIM environment 228
installation
advanced 211
terminals (ASCII)
setting communications options 58, 331
timeline for scripts execution 393
tmp resource 257
defined 257
defining 257
overview 257
troubleshooting 95
boot problems 91
introduction 91
procedure 92
cleaning up failed optional software installation
introduction 23, 345
full /usr file system 94
installation from system backup (mksysb) 87
resolving reported problems 89
introduction 87, 288
network boot problem 313
client and server, establishing network communication 313
obtaining the boot image from the server 313
running the boot image on the client 315
nonprompted mode, overriding 92
producing debug output 315
producing debug output from
a network boot image 316
BOS install program 317
troubleshooting *(continued)*
prompted mode, changing to 92
troubleshooting procedures
recovering /etc/niminfo file 195

U

unconfig operation 280
unconfiguring
master 150
Universal Disk Format 325
update operation 281
update_all
install_all_updates command 342
SMIT fast path 339
updateios operation 281
updates, service
explanation of 335
User Specified Installation Location (USIL) 62
user volume group
backing up 327
definition of 318
USIL 62
USIL connector ODM class object 65
Using Activation Engine 33
using the iSCSI configuration menus 66

V

Value 247
Virtual I/O Server using NIM
installing 175, 178
migrating 164
volume groups
accessing 91
introduction 91
procedure 92
backing up 21, 320, 327
nonroot 318
root 95, 318
user 318

W

warning messages
NIM 288
windows
Change Disk(s) Where You Want to Install (BOS) 59
Installing base operating system (BOS) 414
Installing Base Operating System (BOS) 61
WPAR 116, 117, 118
detached
installing and managing software 224
wpar_spec resource 255
defining 255
overview 255

Printed in USA