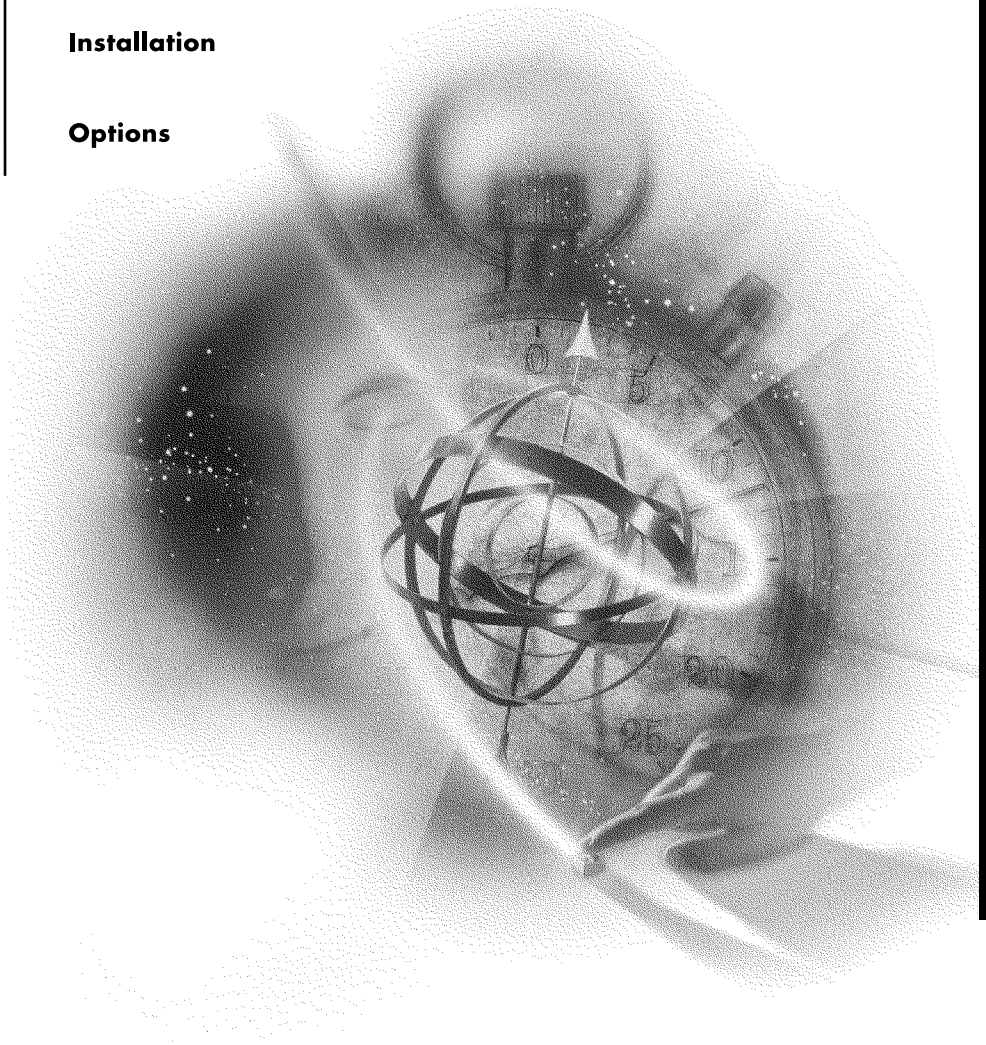
Other Installation Options



NetWare® 5.1

NETWORKING SOFTWARE

Novell®

Legal Notices

Novell, Inc.
122 East 1700 South
Provo, UT 84606
U.S.A.

www.novell.com

Other Installation Options
January 2000
104-001234-001

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

For a list of Novell trademarks, see the final appendix of this book.

Third-Party Trademarks

All third-party trademarks are the property of their respective owners.

Contents

Overview

The following advanced options can help you upgrade or install NetWare® 5.1 servers.

Use NetWare Accelerated Upgrade

You can run NetWare Accelerated Upgrade from a Windows* client workstation, so that you don't need to be physically present at the server console. Although NetWare Accelerated Upgrade is quicker than the standard installation process, it does not install additional network products, licensing services, or license certificates.

Automate the NetWare 5 Installation with a Response File

Installing the NetWare operating system software can be easier and more flexible when you use a response file. When used with the graphical server installation, a response file lets you:

- Set and display specific defaults
- Bypass entire sections of the installation
- Automate the entire server installation process

A response file is a text file containing sections and keys (similar to a Windows .INI file). You can create a response file using any ASCII text editor. If you use a response file, the NetWare server installation reads the installation parameters directly from the response file, replacing the default installation values with response file values.

Use Installation Scripts for NetWare 4 and NetWare 5

NetWare Installation Scripts (formerly known as CDWare Script Installation) let you:

- Alter or extend the NetWare installation process.
- Install additional products or services on a NetWare server after the operating system has been installed.

1 Using NetWare Accelerated Upgrade

The NetWare® Accelerated Upgrade utility is an advanced utility that can be used to upgrade a NetWare 4 or NetWare 5 server to a NetWare 5.1 server. NetWare Accelerated Upgrade is intended for use by system administrators who are skilled at troubleshooting and installing NetWare networks.

IMPORTANT: Back up all data before using this utility. Although NetWare Accelerated Upgrade can reduce the time required to upgrade multiple servers, it must be used carefully. There are no backout procedures that will restore your servers to their original configuration or restore trustee assignments if the upgrade is unsuccessful.

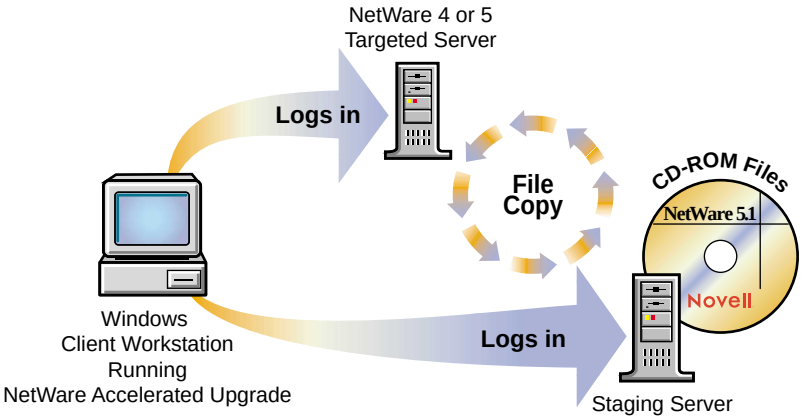
You can run NetWare Accelerated Upgrade from a Windows* client workstation, so that you don't need to be physically be present at the server console.

Although NetWare Accelerated Upgrade is quicker than the standard installation process, it does not install additional network products or licensing certificates.

How NetWare Accelerated Upgrade Works

In order to run, NetWare Accelerated Upgrade requires a Windows client workstation, a staging server, and a NetWare 4 or NetWare 5 server targeted for upgrade.

The staging server holds a copy of the NetWare 5.1 Operating System CD files. From this one location, all targeted servers can be upgraded.



After the NetWare 5.1 Operating System CD files are copied to the staging server, you will log in to a Windows workstation and launch the NetWare Accelerated Upgrade utility (ACCUPG.EXE), located at the root of the NetWare 5.1 Operating System CD. When NetWare Accelerated Upgrade launches, you will log in to the target and staging servers.

Once you have a server-to-server connection, NetWare Accelerated Upgrade compares the existing target server properties, such as the Directory Services version and the CLIB version, with the NetWare 5.1 server requirements for those properties.

Based on the results of the comparison, NetWare Accelerated Upgrade copies the necessary NetWare 5.1 CD files from the staging server to the target server. After the file copy, the target server reboots to complete the NetWare 5.1 server upgrade.

Before you begin upgrading the server, make sure that you have the following software:

- NetWare 5.1 Operating System CD-ROM
- NetWare 5.1 License/Cryptography diskette for a server plus 5 connections

Upgrading to NetWare 5.1

To run NetWare Accelerated Upgrade, you must complete the following tasks:

- Meet system and software requirements
- Prepare a staging server
- Prepare the network
- Prepare the target server
- Run NetWare Accelerated Upgrade
- Complete post-upgrade tasks

Meet System and Software Requirements

NetWare Accelerated Upgrade requires the following system and software configuration.

Requirements for the Windows client workstation:

- ❑ A Windows* 95/98 or Windows NT* workstation with 50 MB of available disk space.
  - The Windows 95/98 workstation must be running Novell® Client™ version 3.1 or later and Support Pack 1 or later.
  - The Windows NT workstation must be running Novell Client for Windows version 4.6 or later and Support Pack 1 or later.
- ❑ PS/2* or serial mouse recommended, but not required.

Requirements for the target server:

- ❑ A server-class PC with Pentium or higher processor with 128 MB of RAM.
- ❑ A target server with the NetWare 4.*x* or NetWare 5 operating system running with the latest updates.

- ❑ (Optional) Load the RSPX and REMOTE NLM™ programs (for the IPX™ protocol) or the RCONJAG6.NLM program (for the IP protocol) on the target server if you want to see and access the target server's server console from the client workstation, eliminating the need to be physically present at the target server.
- ❑ 35 MB of available disk space on the DOS partition.
- ❑ 200 MB minimum available disk space on volume SYS.
- ❑ Make sure that the client workstation, target server, and staging server share the same protocol (IPX or IP).

Table 1 shows the conditions that NetWare Accelerated Upgrade does not support during a NetWare 4.*x* to NetWare 5.1 upgrade. Likewise, Table 2 shows the conditions that NetWare Accelerated Upgrade does not support during a NetWare 5.0 to NetWare 5.1 upgrade.

Table 1 Unsupported procedures during a NetWare 4.*x* to NetWare 5.1 upgrade

Not Supported	Recommendation
The server is running SFT III™	To upgrade, you must break the server mirror and restart the server in native NetWare mode.
The server is running HCSS	Unload any High Capacity Storage System (HCSS) NLM programs that may be running on the server.
Upgrading a NetWare 4.10 server that doesn't have long namespace installed and has over 1 million directory entries	NetWare 4.10 has a limit of 2 million directory entries per volume. If you have more than 1 million directory entries, the NetWare Accelerated Upgrade will fail. During the upgrade, long namespace is installed, which doubles the amount of directory entries in the directory entry table (which may then exceed the 2 million available in NetWare 4.10).

Not Supported	Recommendation
Adding drivers for new network boards and disk drives during the upgrade	Add drivers for network boards and disk drives after upgrading by using NetWare 5.1 utilities, such as NWCONFIG; or, use the NetWare 5.1 Installation program to upgrade.
Adding the IP protocol during the upgrade	Add the IP protocol after upgrading by using NetWare 5.1 utilities, such as INETCFG; or, use the NetWare 5.1 Installation program to upgrade.

Table 2 Unsupported procedures during a NetWare 5.0 to NetWare 5.1 upgrade

Not Supported	Recommendation
Adding drivers for new network boards and disk drives during the upgrade	Add drivers for network boards and disk drives after upgrading by using NetWare 5.1 utilities, such as NWCONFIG; or, use the NetWare 5.1 Installation program to upgrade.
Adding the IP protocol during the upgrade	Add the IP protocol after upgrading by using NetWare 5.1 utilities, such as INETCFG; or, use the NetWare 5.1 Installation program to upgrade.

Continue with "Prepare a Staging Server."

Prepare a Staging Server

The staging server can be any server other than the target servers that you plan to upgrade or the client workstation.

To prepare a staging server:

1. Copy the NetWare 5.1 Operating System CD files to a volume on the staging server. Make sure that all target servers can access the staging server and the client workstation.

   NOTE: You can also mount the NetWare 5.1 Operating System CD as a volume on the staging server.

2 Copy the region-specific cryptography modules.

If you have mounted the NetWare 5.1 Operating System CD as a volume on the staging server, start with Step 2b.

2a At the staging server, create a directory named License at the root of the copied NetWare 5.1 Operating System CD.

2b At the client workstation, insert the NetWare 5.1 License/ Cryptography diskette.

2c Locate the #.NFK file on the NetWare 5.1 License/Cryptography diskette.

The # character represents a variable number. This number is different on every licensing diskette; however, the .NFK extension is always the same.

2d Copy the #.NFK file from the diskette to the newly created License directory at the root of the NetWare 5.1 Operating System CD.

NOTE: If you mounted the NetWare 5.1 Operating System CD, you will need to copy the #.NFK to the SYS:SYSTEM directory on *every* target server that you plan to upgrade.

2e Rename the #.NFK file to NICIFK.

3 (Optional) Specify disk drivers manually.

Although not recommended, you can load disk drivers that are not provided on the NetWare 5.1 Operating System or bypass the hardware auto-detection used while running NetWare Accelerated Upgrade. For more information, see "Specifying Disk Drivers Manually" on page 20.

4 (Optional) Customize the NetWare Accelerated Upgrade script files.

Customizing script files allows you to add additional menu options or install additional software or patches that are not standard features of the NetWare 5.1 installation.

For information on using script files, see Technical Information Document #2944480 at the Novell Support Connection® Web site (http://www.support.novell.com), or see Upgrading an Enterprise Using the NetWare Accelerated Upgrade (AppNotes, April 1999).

NOTE: Customizing the installation process could affect the way that NetWare 5.1 is installed. This may result in an upgrade failure and may make your server unusable.

Continue with "Prepare the Network."

Prepare the Network

Before introducing a NetWare 5.1 server into an existing network containing NetWare 4 or NetWare 5 servers, you must prepare the network by backing up and updating the existing NDS.

To update the network for NetWare 5.1:

1. Log in from a Windows 95/98 or Windows NT workstation to your existing network as a user with Supervisor rights to the container where the server you are upgrading resides.
2. Run the NetWare Deployment Manager utility (NWDEPLOY.EXE), located at the root of the NetWare 5.1 Operating System CD.
3. Complete the following tasks in the Network Preparation section.
   - **3a** Back up NDS® and server data.
   - **3b** Update NDS.
   - **3c** Prepare for NDS 8 (required if installing NetWare 5.1 with NDS 8).
   - **3d** Install or update Novell Licensing Services.

Continue with "Prepare the Target Server."

Prepare the Target Server

To prepare the target server to be upgraded:

1. Update the target server with the latest NetWare software.

   Updates are available at the Novell Support Connection Web site (http://www.support.novell.com). Search for the support pack by entering the name into the Quick Search field.

   For NetWare 4.*x*:

   - NetWare 4.10 requires 410PT8B.EXE, LIBUPI.EXE, STRTL7.EXE and SMSUPG.EXE. Manually copy these files to the SYS:SYSTEM directory. Verify that you are running DS version 5.17 or later.
   - NetWare 4.11 and NetWare 4.2 require NetWare 4 Support Pack 6 (IWSP6A.EXE) or later.

   For NetWare 5.0:

   - NetWare 5.0 requires NetWare 5 Support Pack 2 or later.

2. Run DSREPAIR > Time Synchronization at the server console of the target server. Before continuing, correct all NDS synchronization errors.
3. Mount all volumes.

   NOTE: If you have experienced disk errors in the past, you may want to run VREPAIR on all of the volumes on the server to ensure that there are no errors.
4. Verify that you have server-to-server and server-to-client connections by checking all computer hardware components, such as storage devices and network boards, to verify that they are functioning properly.

   If you have problems here, refer to the troubleshooting section, "Unable to Connect to the Target Server" on page 21.
5. Make sure that you have Supervisor rights to the container where the target server resides.
6. Continue with "Run NetWare Accelerated Upgrade."

Run NetWare Accelerated Upgrade

To run NetWare Accelerated Upgrade:

1. Insert the NetWare 5.1 Operating System CD into the CD-ROM drive of your Windows 95/98 or Windows NT workstation, and log in to your existing network.
2. Run NetWare Accelerated Upgrade (ACCUPG.EXE), located at the root of the NetWare 5.1 Operating System CD.
3. Enter the location where you copied or mounted the NetWare 5.1 CD files.
4. Log in to the target server and make sure that it meets the critical NetWare 5.1 server requirements for the following server properties:
   - NetWare server version
   - Directory Services version
   - CLIB version
   - HCSS is *not* loaded
   - SFT III server is *not* used
   - Unsupported drivers are unloaded

5 Select whether you want NetWare Accelerated Upgrade to update your drivers to NetWare 5.1 drivers, remove any unsupported DOS utilities on a NetWare 4 target server, finish the NetWare 5.1 file copy, and reboot the target server.

NOTE: You may need to manually reboot the target server once the file copy is complete.

When you choose to update your drivers, NetWare Accelerated Upgrade will run hardware auto-detection and replace the existing STARTUP.NCF file with a new STARTUP.NCF file that loads the appropriate .HAM and .CDM drivers that NetWare 5.1 supports. The former startup files will be renamed with a .BAK extension.

IMPORTANT: If the target server contains any disk drivers that are not compatible with NetWare 5.1, they cannot be replaced during the hardware auto-detection process and your server cannot be upgraded.

NOTE: If device drivers are not found for a device currently loaded in the former STARTUP.NCF or AUTOEXEC.NCF files, the LOAD line will be copied to the new file but will be commented out.

Continue with "Complete Post-Upgrading Tasks."

Complete Post-Upgrading Tasks

After you upgrade a server to NetWare 5.1 using the NetWare Accelerated Upgrade utility, the server allows only two connections until you install NetWare 5.1 licenses.

Because the NetWare Accelerated Upgrade utility does not install licenses, use NetWare Administrator to install license certificates on the new NetWare 5.1 server.

NOTE: If licenses are installed at the [Root] of the NDS tree or you are on a MLA account, you may not need to install additional licenses.

1 Install licensing certificates using NetWare Administrator.

1a Log in to your server from the Windows workstation.

1b Run NetWare Administrator (SYS:PUBLIC\WIN32\NWADMN32.EXE).

1c Click Tools > Novell Licensing Services > Add Licenses.

1d Click the License File option.

1e Insert the NetWare 5.1 License/Cryptography diskette into the target server's diskette drive.

1f Follow the on-screen instructions.

If you see a window concerning an Activation Key, see the online help for more information.

NOTE: These connection license certificates require a server assignment. When using NetWare Administrator, you need to set the server assignments manually.

2 Make sure that your existing network products function with NetWare 5.1.

If they are not functioning properly, or if you want to add additional network products, go to the Post-Installation Tasks section of the NetWare Deployment Manager utility (NWDEPLOY.EXE) and select Install NetWare 5.1 Products.

You can also install additional network products through the server console by clicking the red Novell button on the servertop.

Specifying Disk Drivers Manually

You may want to change the behavior of hardware detection and the NDS database upgrade. By removing the -S option from the lines that load HDETECT.NLM and NDSDIBUP.NLM in the HWDETECT.ICS script file, you can confirm or add new hardware devices during the upgrade.

NOTE: To confirm or add new hardware devices during the upgrade, you must physically be at the server console of the target server.

This procedure can also be used to load disk drivers that are not provided on the NetWare 5.1 Operating System CD-ROM. This action may be beneficial when upgrading multiple servers with identical hardware components. If you select specific drivers manually, you must create and configure a \STARTUP directory that contains the disk drivers and the STARTUP.NCF file.

To specify disk drivers manually:

1 Copy the NetWare 5.1 Operating System CD files to a volume on a source server.

IMPORTANT: Do not copy the NetWare 5.1 Operating System CD files onto the target server that you are upgrading or onto the client workstation.

2 On the staging server, create a \STARTUP directory in the same directory where you copied the NetWare 5.1 Operating System CD-ROM files.

3. Create an updated STARTUP.NCF file and copy it to the \STARTUP directory.
4. Copy the appropriate .HAM and .CDM disk drivers to the \STARTUP directory.
5. Continue with "Prepare the Network" on page 17.

Troubleshooting

Unable to Connect to the Target Server

During NetWare Accelerated Upgrade, if you could not make a server-to-server connection, correcting one or more of the following conditions might solve the problem.

Disable SAP Filtering between the Staging and Target Servers

To disable SAP filtering between the staging and target server:

1. At the server console of the target server, enter **INETCFG**.
2. Select Protocols > IPX > Filtering Support.

If SAP filtering cannot be disabled, make sure that the client workstation is on the same LAN segment as the target and staging servers.

Set the Target Server's Gate Parameter to Its Default Router IP Address

If your target server was a NetWare 4 server, complete these steps to set the target server's gate parameter:

1. At the server console, enter **LOAD INSTALL**.
2. Select NCF Files Options > Edit AUTOEXEC.NCF File.
3. In the AUTOEXEC.NCF file, enter a **LOAD TCP/IP** command line.
4. Enter a command line that loads the network board and specifies the slot, frame, and name of the network board (the AUTOEXEC.NCF file should list this information).

   For example, if you had a NE2000™ network board located in slot 3 with an ethernet_II frame, you would enter a command line similar to the following:

   ```
   LOAD NE2000 Slot=3 Frame=ethernet_II Name=NE2000
   ```

5. Enter a command line that will load the network board and its IP address, mask address, and gate parameter (set this to the default router IP address of the target server).

   For example, for the NE2000 network board with a default router IP address of 137.33.172.1, the command line would be similar to:

   ```
   LOAD NE2000 addr=137.65.178.1 mask=255.255.255.0
     gate=137.33.172.1
   ```

6. Enter a command line to bind IP to the network board, using the following format:

   ```
   BIND IP network_board_name
   ```

7. Next, add the IP address, mask address, and gate parameter for the target server to the BIND IP command line.

If your target server was a NetWare 5 server, follow the instructions to set the target server's gate parameter:

1. At the server console, enter **NWCONFIG**.
2. Select NCF Files Options > Edit AUTOEXEC.NCF File.
3. At the BIND IP command line, after the mask address, set the gate parameter to the default router IP address for the target server.

   For example, if your default router IP address was 137.65.178.1, at the BIND IP command line, you would enter the following after the mask address:

   ```
   gate=137.65.178.1
   ```

Detach from Unauthenticated Servers

To detach from an unauthenticated server:

1. From a Windows 95/98 or Windows NT client, right-click the red Novell N.
2. Select NetWare Connections.
3. Select the server that you are not authenticated to.
4. Click Detach.

If you are still having problems communicating with the target server, reboot the client workstation and try to log in to the target server again.

2 Automating the NetWare 5 Installation with a Response File

Installing the NetWare® 5 operating system software can be easier and more flexible when you use a response file. When used with the graphical server installation, a response file lets you:

- Set and display specific defaults.
- Bypass entire sections of the installation.
- Automate the entire server installation process.

A response file is a text file containing sections and keys (similar to a Windows* .INI file). You can create a response file using any ASCII editor.

If you use a response file, the NetWare 5 server installation reads the installation parameters directly from the response file, replacing the default installation values with response file values. The installation program accepts the values as they appear and continues to the next installation screen. Because the installation program gets all the required information from the response file, you can even skip the entire sequence of installation screens.

This chapter contains the following sections:

- “CD Boot and the Response File” on page 79
- “Customizing the Installation Using Install Scripts” on page 80
- “Performing a Factory Install” on page 81

Creating a Response File

You can create a new response file, or edit one from a previous server installation.

NOTE: You should take care when using a response file from a previous installation because it can contain advanced configuration options you don’t want for your current installation.

Creating a New Response File

1. Open an ASCII editor such as EDIT.
2. Input parameters as described in “Syntax” on page 25.

Editing a Response File from a Previous Server Installation

You can generate a response file from a server installation or upgrade, then use it as input for additional server installations or upgrades. Using the response file from another server installation works best when both servers have the same hardware configurations.

1. Perform the first server installation, then reboot the server by clicking Yes on the closing screen of the installation.
2. Log in to the new server after it has been rebooted.
3. Copy RESPONSE.NI from the SYS:NI\DATA directory to either a diskette or to the new server.
4. Modify the file by adjusting the parameters and removing unwanted sections (see “Syntax” on page 25 for a description of the parameters).

Using a Response File

A response file can be used during a server installation in one of two ways:

- Identified during the server installation
- Entered at the command line

Using a Response File During a Server Installation

1. Copy the response file to a diskette, hard disk, or network directory accessible by the server that you are installing NetWare 5 on.
2. Enter **INSTALL** to start the server installation.
3. On the Install Options screen, press F3 for the response file window.
4. Enter the path for the response file, then press Enter.
5. Continue with the rest of the server installation.

Entering a Response File at the Command Line

1. Copy the response file to a diskette, hard disk, or network directory accessible by the server that you are installing NetWare on.
2. Start the server installation from CD, hard disk, or network directory by typing INSTALL /RF=<*response file DOS path\filename*>.

   Replace *response file DOS path* with the path for the response file, and filename with the name of the response file. For example, if you are installing NetWare 5 from a CD and the response file is on a diskette, enter **D:\INSTALL /RF=A:\RESPONSE.TXT**.

Syntax

A response file is a Windows .INI-type file. Data items are identified as keys, and keys have associated values. These keys are grouped in sections. For the NetWare 5 installation program, each data input screen has one or more sections associated with it. Section names of the installation generally correspond to the function of the screens.

Section and key names are case-sensitive. Values associated with keys are not case sensitive. Semicolons placed at the beginning of a line indicate that the line is a comment.

Sections can be placed in any order. However, if there are two or more sections with the same section name, the first section listed in the response file will be used and the other sections will be ignored.

The following sections apply to the NetWare 5 server installation. They are listed in the order they are used during installation.

Section	Purpose
[NWI:Product Information]	Identifies the specific product version the response file is associated with.
[NWI:Language]	Specifies the languages to be installed on the server.
[NWI:Install Options]	Corresponds to the first data input screen.
[NWI:Server Settings]	Corresponds to the server settings screen.
[NWI:Locale]	Corresponds to the regional settings screen.
[NWI:Mouse and Video]	Corresponds to the video and mouse settings screen.
[NWI:Hardware]	Corresponds to the hardware detection and driver matching function of the installation program. If drivers are found in the Update directories, they will be used instead of the drivers found on the NetWare CD.
[NWI:Multi-Processor System]	Identifies the driver associated with the processor system in the server.
[NWI:Storage Adapter n]	Identifies the driver and its associated parameters for a specific storage adapter. This Section can be duplicated for as many adapters as are in the server. n uniquely identifies the sequence of the Section (for example, 1, 2, etc.).

Section	Purpose
[NWI:Storage Device n]	Specifies storage devices. This Section can be duplicated for as many devices as are in the server. *n* uniquely identifies the sequence of the Section (for example, 1, 2, etc.).
[NWI:Network Adapter n]	Identifies the driver and its associated parameters for a specific network adapter. This Section can be duplicated for as many devices as are in the server. *n* uniquely identifies the sequence of the Section (for example, 1, 2, etc.).
[NWI:Reserved Adapter n]	Specifies the adapters that are to be reserved.
[NWI:Hotplug System]	Identifies the driver associated with the processor system in the server.
[NWI:NetWare Loadable Module]	Identifies NLM™ programs that are to be loaded with the hardware drivers.
[NWI:Misc]	Functions as a container of parameters that do not correspond to a data input screen.
[NWI:File System]	Specifies the parameters for the partitions and volumes.
[NWI:File Server]	Specifies the parameters that uniquely identify the server.
[NWI:Protocols]	Controls whether the protocol screen is displayed.
[NWI:TCPIP]	Specifies TCP/IP parameters.
[NWI:IPX]	Specifies IPX™ parameters.
[NWI:IPCMD]	Specifies Compatibility Mode parameters.
[NWI:SNMP]	Specifies Simple Network Management Protocol parameters.

Section	Purpose
[NWI:DNS]	Specifies Domain Name Service (DNS) parameters.
[NWI:Host Name]	Specifies DNS Host Name for each IP address.
[NWI:Time Zone]	Corresponds to the time zone screen.
[NWI:Time Synchronization]	Specifies time server configuration.
[NWI:NDS]	Corresponds to the NDS screen.
[NWI:License]	Identifies the location of the license file.
[NWI:Add to Startup]	Specifies lines to be added to the STARTUP.NCF file.
[NWI:Append To Autoexec.ncf]	Specifies lines to be added to the AUTOEXEC.NCF file.

The Prompt key is listed with each section that corresponds to a data input screen. This key:

- Controls whether the screen will be displayed.
- Lets you use the response file to pass in the keys and values of the section.
- Lets you bypass the screen.

If the Value of Prompt is True, the screen will be displayed and the data that is specified in the response file will be presented as default values. The screen will not be displayed if the Value of Prompt is False. If any of the required data is missing in the response file, however, the screen will be shown regardless of the Prompt value. For more information on each section's keys, see "NetWare 5 Sections and Keys" on page 29.

In the following example, the data input screen will be displayed with the NDS® information already filled in:

```
[NWI:NDS]
Prompt = True
Tree Name = Novell
New Tree = True
Server Context = O=Utah
Admin Context = O=Utah
Admin Login Name = Admin
Admin Password = install
Display Summary = True
```

NetWare 5 Sections and Keys

The NetWare 5 installation sections and their associated keys are specified below. Included with each key are the default value, possible values, whether the Key is required for the NetWare 5 installation to run silently (for example, to create a new NDS tree without any user input, New Tree=True must appear in the response file), an example of the Key, and a short description.

[NWI:Product Information] Section

Major Version = <string>

Default	NetWare 5
Values	(product name)
Key Required	Yes
Example	Major Version = NetWare 5
Purpose	Specifies the name of the product that this response file is associated with. If this value does not match that of the NetWare default response file (normally found in the INSTALL directory on the CD), the installation will not continue.

Minor Version = <strings>

Default	10
Value	0-99
Key Required	Yes
Example	Minor Version = 10
Purpose	Specifies the revision level of the product. If this value does not match that of the NetWare default response file (normally found in the INSTALL directory on the CD image), the installation will not continue.

[NWI:Language] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

Server Language = <integer>

Default	4 (English)
Values	1-100
Key Required	Yes
Example	Server Language = 4
Purpose	Specifies the language that the screens of the installation program will be displayed in and that will be the default language of the server.

Additional Languages = <integers separated by commas>

Default	(none)
Values	1-100 (values are specified in INSTALL\PROFILE.TXT)
Key Required	No
Example	Additional Languages = 5,6
Purpose	Specifies additional languages that will be installed on the server.

[NWI:Install Options] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

Upgrade = <string>

Default	False
Values	True or False
Key Required	Yes
Example	Upgrade = False
Purpose	Specifies whether the installation will be an upgrade or a new installation. False specifies a new installation rather than an upgrade.

Startup Directory = <string>

Default	C:\NWSERVER
Values	(existing DOS path up to 255 characters)
Key Required	Yes
Example	Startup Directory = C:\NWSERVER
Purpose	Identifies the directory where the server's initialization files will reside on the Boot Partition.

Allow User response file = <string>

Default	True
Values	True or False
Key Required	No
Example	Allow User response file = True
Purpose	Makes the F3 key available on the opening screen. It allows a user to input a path to an additional response file. **NOTE:** Values in a user response file will override the values of other response files passed in previously to the installation program.

Abort on Error = <string>

Default	False
Values	True or False
Key Required	No
Example	Abort on Error = True
Purpose	Specifies whether the installation will halt when an error is encountered during an unattended installation. The user will normally see an error screen if this key is set to False.

[NWI:Server Settings] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

NDS Version = <integer>

Default	8
Values	7 or 8
Key Required	Yes
Example	NDS Version = 8
Purpose	Specifies the version of NDS to install on this server.

Load Server at Reboot = <string>

Default	True
Values	True or False
Key Required	No
Example	Load Server at Reboot = True
Purpose	Adds new AUTOEXEC.BAT and CONFIG.SYS files that sets a minimal DOS environment and automatically loads the server. Pre-existing AUTOEXEC.BAT and CONFIG.SYS files on the server will be renamed to *.000. If this value is set to True and AUTOEXEC.000 already exists, then AUTOEXEC.BAT will be renamed to AUTOEXEC.001, etc.

CD Driver = <string>

Default	NetWare
Values	NetWare or DOS
Key Required	No
Example	Source Media Access = NetWare
Purpose	Determines the driver type (DOS or server) used to access the CD during installation.

[NWI:Locale] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

Country Code = <integer>

Default	001
Values	(Values are specified in INSTALL\NLS\<language number>\LOCALE.TXT)
Key Required	Yes
Example	Country Code = 001
Purpose	Specifies the country-oriented settings for display.

Code Page = <integer>

Default	437
Values	(values are specified in INSTALL\NLS\<language number>\LOCALE.TXT)
Key Required	Yes
Example	Code Page = 437
Purpose	Sets the character set for the server.

Keyboard = <string>

Default	United States
Values	(values are specified in INSTALL\NLS\<language number>\LOCALE.TXT)
Key Required	Yes
Example	Keyboard = United States
Purpose	Sets the keyboard type.

[NWI:Mouse and Video] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

Mouse = <string>

Default	PS/2*
Values	PS/2, Serial COM1, Serial COM2, No Mouse (values are specified in INSTALL\NLS\<language number>\LOCALE.TXT)
Key Required	Yes
Example	Mouse = PS/2
Purpose	Specifies the mouse type that will be used for the graphical portion of the installation.

Use Super VGA = <string>

Default	True
Values	True or False
Key Required	No
Example	Use Super VGA = True
Purpose	Indicates whether 256-color Super VGA is set for the graphical portion of the installation. If set to False, 16-color Standard VGA is set.

[NWI:Hardware] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

Update Storage Driver Directory = <DOS path>

Default	(none) The Install always looks in C:\NWUPDATE as well.
Values	(existing DOS path up to 255 characters)
Key Required	No
Example	Update Storage Driver Directory = F:\DRV\UPDATE
Purpose	Specifies the directory path where updated storage (HAM and CDM) drivers will be found; the directory should already exist and be a local DOS path. **NOTE:** Associated DDI files must accompany the storage drivers in this directory.

Update Network Driver Directory = <DOS path>

Default	(none) The Install always looks in C:\NWUPDATE as well.
Values	(existing DOS path up to 255 characters)
Key Required	No
Example	Update Network Driver Directory = F:\DRV\UPDATE
Purpose	Specifies the directory path where updated network (LAN) drivers will be found; the directory should already exist and be a local DOS path. **NOTE:** Associated LDI files must accompany the storage drivers in this directory.

Update PSM Driver Directory = <DOS path>

Default	(none) The Install always looks in C:\NWUPDATE as well
Values	(existing DOS path up to 255 characters)
Key Required	No
Example	Update PSM Driver Directory = F:\DRV\UPDATE
Purpose	Specifies the directory path where updated processor (PSM) drivers will be found; the directory should already exist and be a local DOS path. **NOTE:** Associated PDI files must accompany the storage drivers in this directory.

Update Source Drive Type = <string>

Default	(the type of the install source)
Values	Floppy or CD or Network or Other
Key Required	No, unless one or more of the Update keys above are set.
Example	Update keys above are setUpdate Source Drive Type = Network
Purpose	Specifies the type of all source directories identified in the keys above; all directories must be of the same type.

PSM Detection = <string>

Default	True
Values	True or False
Key Required	No

Example	PSM Detection = True
Purpose	Controls whether driver matching is performed for the processor system. If driver information is identified in the response file (see Multi-Processor System section), detection can be set to False.

Storage Detection = <string>

Default	True
Values	True or False
Key Required	No
Example	Storage Detection = True
Purpose	Controls whether detection of storage adapters and driver matching is performed. If storage adapter and driver information is identified in the response file (see Storage Adapter section), detection can be set to False.

Network Detection = <string>

Default	True
Values	True or False
Key Required	No
Example	Network Detection = True
Purpose	Controls whether detection of network adapters and driver matching is performed. If network adapter and driver information is identified in the response file (see Network Adapter section), detection can be set to False.

Device Detection = <string>

Default	True
Values	True or False
Key Required	No
Example	Device Detection = True
Purpose	Controls whether detection of storage devices and driver matching is performed. If storage adapter and driver information is identified in the response file (see Storage Device section), detection can be set to False.

HotPlug Detection = <string>

Default	True
Values	True or False
Key Required	No
Example	HotPlug Detection = True
Purpose	Controls whether detection of Hot Plug devices and driver matching is performed.

[NWI:Multi-Processor System] Section

Driver File = <filename>

Default	(none)
Values	(name of driver)
Key Required	No, unless PSM Detection=False
Example	Driver File = MPS14.PSM
Purpose	Identifies the driver to use for the server’s multiprocessor system.

[NWI:Storage Adapter n] Section

Example	[NWI:Storage Adapter 1]
Description	This section can be included for each storage adapter in the server. Several Keys are described under this section which are optional; if the driver is detectable by NetWare, only the Driver File key is required for the section. The other Keys and Values are varied and are extracted from the driver's corresponding DDI file.

Driver File = <filename>

Default	(none)
Values	(name of driver)
Key Required	No, unless Storage Detection=False
Example	Driver File = AHA2940.HAM
Purpose	Identifies the driver to use for the storage adapter.

SLOT = <integer>

Default	(none)
Values	(can be a 5-digit Hardware Instance Number assigned on servers with buses that support detection—such as PCI)
Key Required	No, unless the driver is for an ISA adapter
Example	SLOT = 2
Purpose	Specifies the bus slot of the storage adapter to which the driver should be loaded.

INT = <hexadecimal number>

Default	(none)
Values	1-F
Key Required	No, unless the driver is for an ISA adapter.
Example	INT = F
Purpose	Specifies the interrupt that the adapter is set to; this parameter is commonly used for ISA bus adapters.

PORT = <number>

Default	(none)
Values	(3-digit memory address)
Key Required	No, unless the driver is for an ISA adapter
Example	PORT = 170
Purpose	Specifies the memory address the adapter is set to; this parameter is commonly used for ISA bus adapters.

[NWI:Storage Device n] Section

Example	[NWI:Storage Device 1]
Description	This section can be included for each storage device in the server.

Driver File = <filename>

Default	(none)
Values	(name of driver)
Key Required	No, unless Device Detection=False
Example	Driver File = IDECD.CDM

Purpose	Identifies the driver to use for the storage device.

[NWI:Network Adapter n] Section

Example	[NWI:Network Adapter 1]
Description	This section can be included for each network adapter in the server. Several Keys are described under this section which are optional; if the driver is detectable by NetWare, only the Driver File Key is required for the section. The other Keys and Values are varied and are extracted from the driver's corresponding LDI file.

Driver File = <filename>

Default	(none)
Values	(name of driver)
Key Required	No, unless Network Detection=False
Example	Driver File = 3C59X.LAN
Purpose	Identifies the driver to use for the network adapter.

SLOT = <integer>

Default	(none)
Values	(can be a 5-digit Hardware Instance Number assigned on servers with buses that support detection—such as PCI)
Key Required	No, unless the driver is for an ISA adapter
Example	SLOT = 10001

Purpose	Specifies the bus slot of the network adapter to which the driver should be loaded.

INT = <hexadecimal number>

Default	(none)
Values	1-F
Key Required	No, unless the driver is for an ISA adapter
Example	INT = F
Purpose	Specifies the interrupt that the adapter is set to; this parameter is commonly used for ISA bus adapters.

PORT = <number>

Default	(none)
Values	(3-digit memory address)
Key Required	No, unless the driver is for an ISA adapter
Example	PORT = 170
Purpose	Specifies the memory address the adapter is set to; this parameter is commonly used for ISA bus adapters.

Adapter Name = <string>

Default	(none)
Values	(text up to 255 characters)
Key Required	No
Example	Adapter Name = 3C59X_1
Purpose	Specifies a unique name for this adapter.

Frame Type n = <frame type>

Default	(none)
Values	(frame types supported by the driver are found in the driver's LDI file)
Key Required	No
Example	Frame Type 1 = Ethernet_802.2
Purpose	Identifies the network frame type to be used when loading the driver. There can be as many descriptions as there are frame types supported by the driver—*n* uniquely identifies the frame type (for example, 1, 2, etc.).

Logical Name n = <string>

Default	(none)
Values	(text combining the Frame Type with the Adapter Name)
Key Required	No
Example	Logical Name 1 = 3C59X_1_E82
Purpose	Identifies the name to be used when binding a protocol to a frame type on this adapter. There must be a logical name for each frame type loaded with the driver—*n* uniquely identifies the frame type (for example, 1, 2, etc.).

Octet Bit Order = <string>

Default	Purpose
Values	LSB or MSB
Key Required	No
Example	Octet Bit Order = LSB

Purpose	Specifies the bit order (Least Significant Bit or Most Significant Bit) that this network adapter uses for deciphering the node address.

[NWI:Reserved Adapter n] Section

Example	[NWI:Reserved Adapter 1]
Description	This section can be included for each slot in the server.

SLOT = <integer>

Default	(none)
Values	(can be a 5-digit Hardware Instance Number assigned on servers with buses that support detection—such as PCI)
Key Required	No, unless the driver is for an ISA adapter
Example	SLOT = 5
Purpose	Specifies the bus slot that should be reserved. If an adapter is found in this slot, an error is generated.

[NWI:Hotplug System] Section

Driver File = <filename>

Default	(none)
Values	(name of driver)
Key Required	No, unless HotPlug Detection=False
Example	Driver File = CPQSBD.NLM
Purpose	Identifies the driver to use for the Hot Plug system.

[NWI:NetWare Loadable Module] Section

Driver File = <filename>

Default	(none)
Values	(name of NLM)
Key Required	No
Example	Driver File = ROUTE.NLM
Purpose	Identifies the NLM to load during installation. This is typically used for network support NLM programs.

[NWI:File System] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the character-based SYS Volume and Partition screen is displayed.

Allow Volume Properties = <string>

Default	True
Values	True or False
Key Required	No
Example	Allow Volume Properties = True
Purpose	Controls whether the F3 Volume Properties option is displayed.

GUI Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	GUI Prompt = True
Purpose	Controls whether the File System screen is displayed during the graphical portion of the install.

[NWI:Partition n] Section

Example	[NWI:Partition 1]
Description	This section is included for each partition to be defined during installation. Note that the partition on which volume SYS is created must be Partition 0.

Device Type = <string>

Default	(none)
Values	IDE or SCSI
Key Required	Yes
Example	Device Type = IDE
Purpose	Identifies the type of the device the partition is to be placed on.

IDE Device Serial Number = <string>

Default	(none)
Values	(string of characters)
Key Required	No
Example	IDE Device Serial Number = NOVL-111111111111

Purpose	Specifies the serial number of the device designated by the manufacturer.

IDE Device Channel = <integer>

Default	(none)
Values	1 or 2 or 3 or 4
Key Required	No, unless IDE Device Serial Number is not identified and there is more than one IDE drive.
Example	IDE Device Channel = 2
Purpose	Identifies the channel on which the device is to be found

IDE Device Type = <string>

Default	(none)
Values	Master or Slave
Key Required	No, unless IDE Device Serial Number is not identified
Example	IDE Device Type = Master
Purpose	Specifies the IDE type of the device on the channel.

SCSI Device Adapter Slot = <integer>

Default	(none)
Values	1 to16
Key Required	No, unless there is more than one adapter in the server
Example	SCSI Device Adapter Slot = 1
Purpose	Identifies the physical slot number of the SCSI adapter

SCSI Device ID = <integer>

Default	(none)
Values	0 to 15
Key Required	Yes
Example	SCSI Device ID = 0
Purpose	Specifies the ID of the device on the SCSI chain

Partition Type = <string>

Default	NSS
Values	NetWare or NSS or Free
Key Required	Yes
Example	Partition Type = NSS
Purpose	Identifies the type of partition to be created

Partition Size = <integer>

Default	(none)
Values	(up to the size of free space on the device in MB)
Key Required	No
Example	Partition Size = 1000
Purpose	Identifies the size of the partition. Note that the number is modified to align with a cylinder boundary.

Partition HotFix Size = <integer>

Default	(calculated on Partition Size)
Values	0 to 120 (in MB)
Key Required	No

Example	Partition HotFix Size = 20
Purpose	Identifies the size of the redirection area on the partition

Mirror Partition = <integer>

Default	(none)
Values	(partition number)
Key Required	No
Example	Mirror Partition = 3
Purpose	Identifies the partition that this partition will be mirrored to.

[NWI:Volume n] Section

Example	[NWI:Volume 1]
Description	This section can be included for each volume defined on the server. Note that SYS must be Volume 0.

Volume Name = <string>

Default	(none)
Values	(valid volume name)
Key Required	Yes
Example	Volume Name = DATA
Purpose	Identifies the name of the volume; follows naming convention found in the NetWare 5 user documentation.

Volume Type = <string>

Default	NSS
Values	NetWare or NSS or Free

Key Required	Yes
Example	Volume Type = NSS
Purpose	Specifies the type of the volume to be created. Note that volume types must be created on like partition types (e.g., NSS volume on NSS partition).

Block Size = <integer>

Default	(calculated on the size of the first segment of the volume)
Values	4 or 8 or 16 or 32 or 64
Key Required	No
Example	Block Size = 64
Purpose	Identifies the volume's block size. Note that this key does not apply to NSS volumes.

Compression = <string>

Default	True
Values	True or False
Key Required	No
Example	Compression = True
Purpose	Specifies whether compression is enabled on the volume.

Suballocation = <string>

Default	True (on NetWare volume types)
Values	True or False
Key Required	No
Example	Suballocation = True

Purpose	Specifies whether suballocation is enabled on the volume. Note that this key does not apply to NSS volumes.

Data Migration = <string>

Default	False
Values	True or False
Key Required	No
Example	Data Migration = True
Purpose	Specifies whether data migration is enabled on the volume. Note that this key does not apply to NSS volumes.

Mount = <string>

Default	False
Values	True or False
Key Required	No
Example	Mount = True
Purpose	Specifies whether the volume should be mounted during installation. This is useful if additional products and services are to be installed on a volume other than SYS. Note that mounting volumes will take additional time during the server installation. Also, this key will be ignored if GUI Prompt = False is not set in the NWI:File System section; this is to avoid a conflict between this key and the mount volumes screen during installation.

Segment n Partition = <integer>

Default	(none)
Values	1 to 8 (limit for NetWare partitions)

Key Required	Yes
Example	Segment 1 Partition = 1
Purpose	Identifies the partition on which the volume segment is to be created.

Segment n Size = <integer>

Default	(none)
Values	(up to free space on a partition in MB)
Key Required	Yes
Example	Segment 1 Size = 1000
Purpose	Specifies the size of the volume segment.

[NWI:Misc] Section

Relogin Password = <string>

Default	(none)
Values	(text)
Key Required	No, unless it is a network installation.
Example	Relogin Password = Novell
Purpose	Specifies the password for the user connection being used while installing across the network. When this section is properly used, the install bypasses the reauthentication screen following the loading of the LAN driver and prior to the preliminary file copy.

[NWI:File Server] Section

NOTE: This is the first section used during the graphical portion of the installation.

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

Servername = <string>

Default	(none)
Values	(text up to 47 characters—see NetWare documentation for valid set of characters)
Key Required	Yes
Example	Servername = Novell
Purpose	Specifies the name of the server. Note that a server is brought up with temporary name until a valid Servername is entered.

Server ID Number = <hexadecimal number>

Default	(randomly generated)
Values	00000001-FFFFFFFF
Key Required	Yes
Example	Server ID Number = 01010101
Purpose	Uniquely identifies the server for network communication purposes. Note that this number should not conflict with any existing IPX addresses on your network.

[NWI:Protocols] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the protocols screen is displayed.

[NWI:TCPIP] Section

Logical Name n = <string>

Default	(none)
Values	(name identified in the Network Adapter sections of the response file)
Key Required	No, unless IP is to be bound.
Example	Logical Name 1 = 3C59X_1_EII
Purpose	Identifies the name of the adapter and frame type that the protocol is to be bound to.

IP Address n = <decimal number>

Default	(none)
Values	(valid IP address)
Key Required	No, unless Logical Name is set
Example	IP Address 1 = 130.1.130.1
Purpose	Specifies the IP address to be bound.

Subnet Mask n = <decimal number>

Default	(none)
Values	(valid IP address)
Key Required	No, unless Logical Name is set.
Example	Subnet Mask 1 = 255.255.255.0
Purpose	Specifies the subnet. Divides the IP address into network address and node address.

Gateway n = <decimal number>

Default	(none)
Values	(valid IP address)
Key Required	No
	Gateway 1 = 127.127.0.254
Purpose	Identifies the gateway or router through which this server can communicate outside the subnet.

[NWI:IPX] Section

Logical Name n = <string>

Default	(none)
Values	(name identified in the Network Adapter sections of the response file)
Key Required	No, unless IPX is to be bound
Example	Logical Name 1 = 3C59X_1_E82
Purpose	Identifies the name of the adapter and frame type that the protocol is to be bound to.

IPX Address n = <hexadecimal number>

Default	(address detected by the installation program)
Values	00000001-FFFFFFFF
Key Required	No, unless Logical Name is set
Example	IPX Address 1 = 01010102
Purpose	Specifies the IPX address to be bound.

[NWI:IPCMD] Section

IPX Compatibility = <string>

Default	True
Values	True or False
Key Required	No
Example	IPX Compatibility = True
Purpose	Controls whether SCMD is loaded on this server.

Migration Agent = <string>

Default	True
Values	True or False
Key Required	No
Example	Migration Agent = True
Purpose	Controls whether the Migration Agent (i.e., SCMD /MA) is loaded on this server.

CMD Net Number = <string>

Default	(none)
Values	00000001-FFFFFFFF

Key Required	No
Example	CMD Net Number = 01010101
Purpose	Specifies the network number.

Preferred IP Address = <decimal number>

Default	(none)
Values	(valid IP address)
Key Required	No
Example	Preferred IP Address = 130.1.130.1
Purpose	Specifies the IP address.

[NWI:SNMP] Section

Hardware = <string>

Default	(none)
Values	Values (text)
Key Required	No
Example	Hardware = Personal Computer
Purpose	Identification information that is sent with SNMP traps to assist in troubleshooting.

Location = <string>

Default	(none)
Values	(text)
Key Required	No
Example	Location = Building A
Purpose	Identification information that is sent with SNMP traps to assist in troubleshooting.

Contact = <string>

Default	(none)
Values	(text)
Key Required	No
Example	Contact = John Doe
Purpose	Identification information that is sent with SNMP traps to assist in troubleshooting.

Trap Targets TCPIP = <decimal numbers separated by commas>

Default	(none)
Values	(valid IP addresses)
Key Required	No
Example	Trap Targets TCPIP = 127.1.127.3
Purpose	Identifies IP address of a computer to which SNMP traps are sent.

Trap Targets IPX = <MAC address separated by commas>

Default	(none)
Values	(8-character IPX address:12-character node number)
Key Required	No
Example	Trap Targets IPX = C9990111:000001B555555
Purpose	Identifies MAC addresses of a computer to which SNMP traps are sent.

[NWI:DNS] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the DNS screen is displayed.

Domain = <string>

Default	(none)
Values	(text)
Key Required	No
Example	Domain = novell.com
Purpose	Specify a domain.

Nameservers = <decimal numbers separated by commas>

Default	(none)
Values	(valid IP addresses)
Key Required	No
Example	Nameservers = 131.1.131.1, 131.1.131.2
Purpose	Identifies the servers to be used for name resolution.

[NWI:Host Name] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the screen is displayed.

Host Name n = <string>

Default	(none)
Values	(text)
Key Required	No
Example	Host Name 1 = install.novell.com
Purpose	Specifies the host name associated with an IP address already bound.

IP Address n = <decimal number>

Default	(none)
Values	(valid IP address)
Key Required	No
Example	IP Address 1 = 130.1.130.1
Purpose	Specifies the IP address associated with a Host Name. Note that Host Name 1 will be set as the Primary address.

[NWI:Time Zone] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the time zone screen is displayed.

Time Zone = <string>

Default	(none)
Values	(string—see COMOTHR.ILS)
Key Required	Yes
Example	Time Zone = (GMT-06:00) U.S. and Canada Central Time
Purpose	Specifies the standard time zone. Note that the exact string from COMOTHR.ILS (in INSTALL\NLS\4) must be used.

Use Daylight Saving Time = <string>

Default	True, if Time Zone supports Daylight Saving Time
Values	True or False
Key Required	No
Example	Use Daylight Saving Time = True
Purpose	Controls whether daylight saving time is set.

[NWI:Time Synchronization] Section

Default Time Server Type = <string>

Default	(Single if new tree, Secondary if existing tree)
Values	Single, Primary, Secondary
Key Required	No
Example	Default Time Server Type = Single
Purpose	Specifies the time server type.

[NWI:NDS] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the NDS screen is displayed.

Tree Name = <string>

Default	(none)
Values	(text)
Key Required	Yes
Example	Tree Name = Novell
Purpose	Specifies the tree name for either a new tree or an existing tree to install the server into.

New Tree = <string>

Default	False
Values	True or False
Key Required	Yes
Example	New Tree = True
Purpose	Specifies whether the NDS tree to be installed is a new tree or install into an existing tree. **NOTE:** This key is only used when Prompt = False.

Server Context = <string>

Default	(none)
Values	(NDS distinguished name)
Key Required	Yes
Example	Server Context = .install.novell
Purpose	Identifies the NDS context that the server is to be installed into.

Add Replica = <string>

Default	(none)
Values	True or False
Key Required	No
Example	Add Replica = True
Purpose	Controls whether a replica is added to the server. This only applies if the server is not one of the first three servers added to an NDS tree (by default the first three servers receive replicas).

Upgrade User Conflict Action = <string>

Default	(none)
Values	Delete, Merge, or Rename
Key Required	No, unless it is an upgrade from NetWare 3.1*x*
Example	Upgrade User Conflict Action = Merge
Purpose	Determines what action is to be taken during the upgrade of a NetWare 3.1*x* server when objects of the same name are found during the conversion of the bindery to NDS.

Admin Login Name = <string>

Default	(none)
Values	(text)
Key Required	Yes
Example	Admin Login Name = Admin
Purpose	Specifies the name for the server administrator.

Admin Context = <string>

Default	(none)
Values	(NDS distinguished name)
Key Required	Yes
Example	Admin Context = .install.novell
Purpose	Identifies the NDS context in which the server administrator is to be installed into.

Admin Password = <string>

Default	(none)
Values	(text)
Key Required	Yes
Example	Admin Password = Novell
Purpose	Specifies the password for the server administrator. Note that for security purposes, this password is removed from the response file after NDS is installed.

Admin Language = <integer>

Default	4 (English)
Values	1-100
Key Required	Yes
Example	Admin Language = 4
Purpose	Specifies the language of the server administrator. User objects created by the server administrator will be assigned this language.

Display Summary = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Display Summary = True
Purpose	Controls whether the NDS Summary screen is displayed

[NWI:License] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the license screen is displayed.

Display License Agreement = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Display License Agreement = False
Purpose	Controls whether the license agreement screen at the beginning of the server install is displayed

Install Licenses Later = <string>

Default	False
Values	True or False
Key Required	No
Example	Install Licenses Later = True
Purpose	Specifies that no licenses will be installed on the server. If this key is set to True, all remaining licensing keys will be ignored.

License File = <string>

Default	(none)
Values	(existing DOS path and envelope file name—up to 255 characters)
Key Required	No, unless a license is to be installed
Example	License File = C:\NWSERVER\0000001.NLF
Purpose	Identifies the directory and filename of a license envelope file.

Search For Existing License = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Search For Existing License = False
Purpose	Controls whether the licenses (such as MLA licenses) are searched for in the tree. Used in conjunction with the next key.

Auto Skip If License Found = <string>

Default	False
Values	True or False
Key Required	Yes
Example	Auto Skip If License Found = True
Purpose	Controls whether the license that was found using the previous key is used; this is particularly useful for MLA accounts.

[NWI:Add to Startup] Section

Prepend Line n = <decimal number>

Default	(none)
Values	(decimal number)
Key Required	No
Example	Prepend Line 1 = SET IGNORE DISK GEOMETRY=ON
Purpose	Adds the line to the beginning of the STARTUP.NCF file. If multiple lines are listed, *n* should increment sequentially.

Append Line n = <decimal number>

Default	(none)
Values	(decimal number)
Key Required	No
Example	Append Line 1 = LOAD NFS.NAM
Purpose	Adds the line to the end of the STARTUP.NCF file. If multiple lines are listed, *n* should increment sequentially.

[NWI:Append To Autoexec.ncf] Section

Linen = <decimal number>

Default	(none)
Values	(decimal number)
Key Required	No
Example	Line 1 = LOAD MONITOR
Purpose	Adds the line to the end of the AUTOEXEC.NCF file. If multiple lines are listed, *n* should increment sequentially.

Perform a Fully Automated Installation

The following parameters must be present to completely automate an installation or upgrade. The syntax and values for the supplementary NetWare 5 installation sections and their associated keys are specified below. For a fully automated installation to work, these sections must appear in the response file in addition to all of the required keys listed in "NetWare 5 Sections and Keys" on page 29.

[Initialization] Section

SummaryPrompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	SummaryPrompt = False
Purpose	Controls whether the installation Summary screen is displayed.

[Novell:NOVELL_ROOT:1.0.0] Section

closeScreen = <string>

Default	(none)
Values	SilentCloseScreen
Key Required	Yes
Example	closeScreen = SilentCloseScreen
Purpose	When this parameter is set, the final installation screen is not shown.

Reboot = <string>

Default	(none)
Values	True or False
Key Required	Yes

Example	Reboot = False
Purpose	Controls whether the server is restarted after the final screen of the installation is closed.

Automate the Installation of Additional Products and Services

As part of the overall NetWare 5 installation, you can auto-install additional products and services by including a [Selected Nodes] section in the response file. Because this section of the response file is not as easy to configure as the other sections are, we recommended the following:

1. Perform a manual installation, selecting the products you want to have installed on the server.
2. Modify the resulting response file for use as input to future NetWare 5 installations.
3. Add a [Selected Nodes] section to the input response file.
4. Add the following lines under the [Selected Nodes] section heading:

```
Novell:NetWare5:1.0.0=Novell:NetWare5OS:5.0.0,Novell:Products:1.0.0,Novell:NW
UpdateGroup:1.0.0
Novell:NetWare5OS:5.0.0=Novell:DiskCarver:1.0.0,Novell:Protocols:1.0.0,Novell
:DS_Install:1.0.0,      Novell:LicensePrompt:1.0.0,Novell:NW:1.0.0,Novell:NDPS
Server Files:1.0.0
Novell:NW:1.0.0=Novell:Startup:1.0.0,Novell:SYS:1.0.0,Novell:DriverFiles:1.0.
0
Novell:Startup:1.0.0=Novell:StartupDirectory:1.0.0
Novell:SYS:1.0.0=Novell:SYSDirectory:1.0.0,Novell:ETCDirectory:1.0.0,Novell:P
ROFINST_NODE:1.0.0
Novell:DriverFiles:1.0.0=Novell:LANFiles:1.0.0,Novell:SBDFiles:1.0.0
Novell:NDPS     Server     Files:1.0.0=Novell:NDPS     System:1.0.0,Novell:NDPS
Public:1.0.0
Novell:Products:1.0.0=Novell:NICIInstall:1.0.0
Novell:NICIInstall:1.0.0=Novell:NICIModule:1.0.0
Novell:NWUpdateGroup:1.0.0=Novell:NWUpdate:1.0.0
```

NOTE: You can add more lines according to your needs, but these represent the minimum information that must be included in the [Selected Nodes] section.

[Selected Nodes] Section

Prompt = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Prompt = True
Purpose	Controls whether the Additional Products and Services screen is displayed.

Product Selection = <string>

Default	(none)
Values	Default or Web
Key Required	No
Example	Product Selection = Web
Purpose	Specifies which bundle of Additional Products and Services to install. Default will install the Web Services bundle. Web will install the Web Services bundle plus WebSphere* Application Server. To specify a custom bundle of Additional Products and Services, do not include this key, and specify each product as described below.

Novell:Products:1.0.0 = <strings separated by commas>

Default	Novell:NICIInstall:1.0.0 **NOTE:** This NICI install string must be included.

Values	Novell:NDPS:2.0.0 Novell:IpLdapService:3.0.0 Novell:IpCatalogService:1.0.0 Novell:IpWanmanService:1.0.0 Novell:SecuritySASInstall:1.0.0 Novell:PKIInstall:1.0.0 Novell:NICIInstall:1.0.0 Novell:RAS:4.1.0 Novell:SMS:1.0.0 Novell:DNS_DHCP:1.0.0
Key Required	Yes
Example	Novell:Products:1.0.0=Novell:NICIInstall:1.0.0,Novell:NDPS:2.0.0
Purpose	Specifies the additional products and services to be installed on the server.

Installing Multiple Products on a Server

To install multiple products on a server, make sure that the Novell:Products:1.0.0 key has multiple values, separated by commas as shown in the example above.

Product Keys and Values

In addition to identifying the product in the Novell:Products:1.0.0 key, each product has its own keys and values for the [Selected Nodes] section. These keys and values for each product are listed below. We recommended that you reuse a response file from a previous installation rather than attempting to type these keys and values into the file. If you do not enter them exactly as shown, the NetWare 5 Installation will fail abruptly with no information to help you troubleshoot the problem.

Novell Distributed Print Services (NDPS)

```
Novell:NDPS:2.0.0=Novell:NDPS    Server    Files:1.0.0,Novell:NDPS    Resource
Files:1.0.0
Novell:NDPS    Resource    Files:1.0.0=Novell:NDPS    Banner:1.0.0,Novell:NDPS
Font:1.0.0,Novell:NDPS Prndef:1.0.0,Novell:NDPS Prndrv:1.0.0
Novell:NDPS   Prndrv:1.0.0=Novell:NDPS   Prndrv   W31:1.0.0,Novell:NDPS   Prndrv
W95:1.0.0,Novell:NDPS Prndrv NT4:1.0.0
```

LDAP Services

```
Novell:SecuritySASModule:1.0.0=Novell:SASFiles:1.0.0
Novell:PKIModule:1.0.0=Novell:PKIFiles:1.0.0
```

NDS Catalog Services

```
Novell:IpCatalogService:1.0.0=Novell:DfgCatalogService:1.0.0
```

WAN Traffic Manager Services

```
Novell:IpWanmanService:1.0.0=Novell:DfgWanmanService:1.0.0
```

Secure Authentication Services

```
Novell:SecuritySASInstall:1.0.0=Novell:SecuritySASModule:1.0.0,Novell:PKIModu
le:1.0.0,Novell:NICIModule:1.0.0
Novell:SecuritySASModule:1.0.0=Novell:SASFiles:1.0.0
Novell:PKIModule:1.0.0=Novell:PKIFiles:1.0.0
```

Novell PKI Services

```
Novell:PKIInstall:1.0.0=Novell:PKIModule:1.0.0,Novell:NICIModule:1.0.0
Novell:PKIModule:1.0.0=Novell:PKIFiles:1.0.0
```

Novell Internet Access Server

```
Novell:RAS:4.1.0=Novell:RAS                                          Server
Files:1.0.0,Novell:SVCDEF_NODE:1.0.0,Novell:SupportedOS:1.0.0,Novell:UpgradeF
rom:1.0.0
Novell:RAS Server Files:1.0.0=Novell:RAS System Connect:1.0.0,Novell:RAS System
Connect Scripts:1.0.0,Novell:RAS Public:1.0.0
```

Storage Management Services

```
Novell:SMS:1.0.0=Novell:SMSFiles:1.0.0
Novell:SMSFiles:1.0.0=Novell:SMSSystemFiles:1.0.0,Novell:SMSPublicFiles:1.0.0
```

Novell DNS/DHCP Services

No additional keys and values are used for the installation of this service.

Additional Services

Two of the additional services available with NetWare 5, LDAP Services and Novell® DNS/DHCP Services, have data input screens for their configuration. These services have response file sections that must be included in addition to the keys and values mentioned above.

NOTE: Be sure to use the proper case (upper or lower) when entering any of these keys and values.

[LDAP] Section

prompt = <string>

Default	(none)
Values	true or false
Key Required	Yes
Example	prompt=false
Purpose	Controls whether the LDAP configuration screen is displayed.

adminID = <string>

Default	(none)
Values	(NDS distinguished name)
Key Required	Yes
Example	adminID=.CN=admin.O=install
Purpose	Identifies the Admin name and NDS context; this should correspond with the Admin Login Name and Admin Context identified in the NWI:NDS section of the response file. Note the case of the characters in the string.

installCatalog = <string>

Default	(none)
Values	true or false

Key Required	Yes
Example	installCatalog=false
Purpose	Controls whether an LDAP catalog is installed. Corresponds to the question: Enable use of LDAP catalog?

useCatalogOnly = <string>

Default	(none)
Values	true or false
Key Required	Yes
Example	useCatalogOnly=false
Purpose	Controls whether the LDAP catalog is used exclusively for searching. Corresponds to the LDAP screen prompt: Catalog Usage for Searching.

[Novell:DNS_DHCP:1.0.0] Section

Prompt = <string>

Default	None
Values	true or false
Key Required	Yes
Example	Prompt=false
Purpose	Controls whether the DNS/DHCP configuration screen is displayed.

TreeName = <string>

Default	(none)
Values	(text)
Key Required	Yes
Example	TreeName=Novell

Purpose	Specifies the NDS tree name in which DNS/DHCP Services will be installed.

UserName = <string>

Default	(none)
Values	(NDS distinguished name)
Key Required	Yes
Example	UserName=.CN=admin.O=install
Purpose	Identifies the Admin name and NDS context; this should correspond with the Admin Login Name and Admin Context identified in the NWI:NDS section of the response file. Note the case of the characters in the string.

ExtendDNIPSchema = <string>

Default	(none)
Values	true or false
Key Required	Yes
Example	ExtendDNIPSchema=true
Purpose	Controls whether the schema is extended for DNS/DHCP Services. Note that this should be set to true.

LocatorNDSContext = <string>

Default	(none)
Values	(NDS distinguished name)
Key Required	Yes
Example	LocatorNDSContext=O=install
Purpose	Identifies the NDS context that the Locator Object is to be installed into.

GroupNDSContext = <string>

Default	(none)
Values	(NDS distinguished name)
Key Required	Yes
Example	GroupNDSContext=O=install
Purpose	Identifies the NDS context that the Group Object is to be installed into.

RootSrvrNDSContext = <string>

Default	(none)
Values	(NDS distinguished name)
Key Required	Yes
Example	RootSrvrNDSContext=O=install
Purpose	Identifies the NDS context that the RootSrvr Zone is to be installed into.

In addition to the DNS/DHCP keys identified above, there must also be a line in the [Settings] section with the entry: Novell:DNS_DHCP:1.0.0=

CD Boot and the Response File

There are two options available for passing a response file into the NetWare 5 installation: using the /RF switch at the command line, or from the Install Options screen.

If you start the install by booting directly to the NetWare 5 CD, you cannot pass in the response file automatically unless you have a RESPONSE.TXT file in the C:\NWUPDATE directory. When the NetWare 5 CD is booted, the startup utility checks for a RESPONSE.TXT file in the C:\NWUPDATE directory. If such a file exists, the installation program bypasses the DOS partitioning utility and uses RESPONSE.TXT as the input response file.

Customizing the Installation Using Install Scripts

With NetWare 5, install scripts are supported by the NWCONFIG utility and the NetWare 5 installation system. There are two places in the NetWare 5 installation where you can use install scripts:

- At the end of the preliminary file copy (just prior to the launching of the graphical portion of the install). An install script executed here can be used for copying files from the DOS partition to the SYS volume.
- At the end of the NetWare 5 installation. It is called right after you answer AOK@ or ANo@ on the closing screen and before the install cleanup process. This script is useful if you want to manage files and launch NLM programs (for example, those that install other products) as part of the NetWare 5 installation.

The following response file syntax is used to run install scripts during the NetWare 5 installation:

[NWI:Install Script] Section

Script Location = <string>

Default	None
Values	(existing DOS path up to 255 characters)
Key Required	No
Example	Script Location = C:\NWUPDATE\PRECOPY.IC?
Purpose	Identifies the filename and path of an Install Script that will be executed at the end of the preliminary file copy (prior to the launching of the graphical portion of the install). **NOTE:** Usage of the ? wildcard in the extension of the filename will suppress an error if the specified file does not exist.

Close Script = <string>

Default	None
Values	(existing DOS path up to 255 characters)
Key Required	No
Example	Script Location = SYS:\OTHER.ICS
Purpose	Identifies the filename and path of an Install Script that will be executed when the final screen of the NetWare 5 server installation is closed. **NOTE:** This script is executed just prior to some housekeeping that is done by the Close Screen routine.

Performing a Factory Install

The AFactory Install can reduce the complexity and time spent on an installation. Much of the hardware configuration and file copy is done in the AFactory so that you only need to use the graphical portion of the NetWare 5 installation to complete the on-site configuration. Because the network operating system files are pre-copied to the server, installation time is reduced to a few minutes.

A Factory Install splits the installation of NetWare into two phases:

- Phase 1 performs the disk detection, disk partitioning, volume SYS creation, and file copy portions of the install. This phase is meant to be performed in a factory or configuration center.
- Phase 2 focuses on user configuration. It includes the setting of the server name, protocol binding, creation of volumes other than SYS, and configuration of time zone, NDS, licensing, and other products. This phase is meant to be performed by the customer at the server's permanent location.

Once Phase 1 is completed, the machine should be rebooted or powered off. When the machine is powered on and the server is started, the AUTOEXEC.NCF file will launch Java* for Phase 2 of the installation.

A Factory Install is implemented with the response file sections below. The Preinstall key of the [NWI:Factory] section is the primary key that directs the NetWare 5 Installation to perform a factory install. The Precopy key of the other sections specifies whether the file group should be recopied during the final file copy routine of the NetWare 5 Installation. If Precopy=True, the files will be verified, but not copied. The Installation will appear, however, as if it is copying the files because you will see each file name displayed in the copy status box.

NOTE: The NetWare 5 CD must be inserted in Phase 2 of the Factory Install in order to perform this file verification.

All other required keys for an automated install must also be included in the response file. See "Perform a Fully Automated Installation" on page 71 for more information.

[NWI:Factory] Section

Preinstall = <string>

Default	True
Values	True or False
Key Required	Yes
Example	Preinstall = True
Purpose	Controls whether the Factory Install option of the NetWare 5 installation is invoked. If Preinstall=True, the remaining keys are checked.

[Novell:SYSDirectory:1.0.0] Section

Precopy = <string>

Default	True
Values	True or False
Key Required	No
Example	Precopy = True

Purpose	Controls whether the files destined for volume SYS are copied in Phase 2. If Precopy=True, the files are not copied again in the final file copy routine.

[Novell:ETCDirectory:1.0.0] Section

Precopy = <string>

Default	True
Values	True or False
Key Required	No
Example	Precopy = True
Purpose	Controls whether the files destined for the SYS:\ETC directory are copied in Phase 2. If Precopy=True, the files are not copied again in the final file copy routine.

[Novell:LANFiles:1.0.0] Section

Precopy = <string>

Default	True
Values	True or False
Key Required	No
Example	Precopy = True
Purpose	Controls whether the LAN files destined for the C:\<Startup Directory>\DRIVERS directory are copied in Phase 2. If Precopy=True, the files are not copied again in the final file copy routine.

[Novell:StorageFiles:1.0.0] Section

Precopy = <string>

Default	True
Values	True or False
Key Required	No
Example	Precopy = True
Purpose	Controls whether the HAM and CDM files destined for the C:\<Startup Directory>\DRIVERS directory are copied in Phase 2. If Precopy=True, the files are not copied again in the final file copy routine.

[Novell:PSMFiles:1.0.0] Section

Precopy = <string>

Default	True
Values	True or False
Key Required	No
Example	Precopy = True
Purpose	Controls whether the PSM files destined for the C:\<Startup Directory>\DRIVERS directory are copied in Phase 2. If Precopy=True, the files are not copied again in the final file copy routine.

[Novell:StartupDirectory:1.0.0] Section

Precopy = <string>

Default	True
Values	True or False

Key Required	No
Example	Precopy = True
Purpose	Controls whether the files destined for the C:\<Startup Directory> directory are copied in Phase 2. If Precopy=True, the files are not copied again in the final file copy routine.

3 Installation Scripts for NetWare 4 and NetWare 5

This chapter describes the installation script support in NetWare® 4 and NetWare 5. NetWare Installation Scripts (formerly known as CDWare Script Installation) let you:

- Alter or extend the NetWare pre-packaged installation process.
- Install additional products or services onto a NetWare server after the operating system has been installed.

Using NetWare Installation Scripts

NetWare installation scripts are processed by the following NetWare installation NLM™ programs:

- INSTALL.NLM for NetWare 4
- NWCONFIG.NLM for NetWare 5

Script Files

Script files are ASCII text files. They can have any eight-character name with one of the following extensions:

Extension	Type of File
.IPS	Individual product installation
.ICS	Common functionality between scripts or non-language specific scripts
.ILS	Language-specific functionality

Launching a Script File

1. Load INSTALL.NLM (for NetWare 4) or NWCONFIG.NLM (for NetWare 5).
2. Click Products Options on the Configuration Options screen.
3. Click Other Installation Actions menu > Install a product not listed.
4. Press F3, then specify the directory path to the script file.

The directory path is first checked for an .IPS file. If only one .IPS file exists, it is executed. If more than one exists, only the first one found is executed. If there is no .IPS file, PINSTALL.NLM will be loaded and executed. The first @Other...@EndOther descriptor found in the .IPS file is displayed so that you can confirm the product being installed.

The NetWare install program also loads any file that contains the command line parameter "B=". For example, with NetWare 5, a script can be launched by typing LOAD NWCONFIG b=A:\<*myscript*>.IPS (where <*myscript*> stands for the name of your .IPS file).

Using a Script with NetWare 4

NetWare installation scripts control the installation process of NetWare 4. An interpreter first pre-processes the script file, then executes the installation commands.

With NetWare 4, INSTALL.BAT loads SELECT.EXE and NWNSTLL.EXE which starts the script processing of NETMAIN.ILS. The script invokes INSTALL.NLM in batch mode (non-interactively), using separate script files for input and for errors. Both the input and error files are specified on the command line when the installation NLM is loaded.

Below are the basic command line parameters needed to run the installation NLM in batch mode:

Parameter	Description
b	*<batch file spec>*
d	*<destination path in DOS for SERVER.EXE, etc.>*
e	*<error file spec>*

Parameter	Description
NOSTATUS	Suppress status bar display in batch mode
s	*<source path DOS specifier>*

For example:

```
LOAD INSTALL.NLM b=L:\MYDIR\CDSCRPT.BAT, s=C:\DOS,
  d=C:\DOS, e=L:\MYDIR\ERR.FIL, NOSTATUS
```

During installation, NETMAIN.ILS and other associated files are normally copied to the startup or boot directory (the DOS directory where STARTUP.NCF resides). The installation program scans the files when needed, and the information is retained in memory for use in various installation steps. You can modify these NetWare script files to copy additional files or launch additional processes, but we do not recommend it.

For a list of the installation NLM switches, see "Installation NLM Command Line Options" on page 146.

Using a Script with NetWare 5

There are two points in the installation where scripts can be launched:

- At the end of the preliminary file copy just prior to the launching of the graphical portion of the installation.
- At the end of the installation.

See Using a Response File with NetWare 5 for more information.

The installation program in NetWare 5 uses one installation script (NETWARE.ILS) during the text-based file copy. You can modify this script file to copy additional files or launch additional processes, but we do not recommend it.

Descriptors and Commands

This section describes the descriptors and commands found in NETMAIN.ILS and other installation script files. The product installation script files you write contain the same types of descriptors and commands as those in NETMAIN.ILS.

A descriptor is a keyword preceded by the @ character. The installation NLM scans the descriptors and uses them to build linked lists and menus in memory. Descriptors

- Describe a configuration or presentation, but do not specify a procedure or order.
- Are typically block-oriented (with an @Command and an @EndCommand statement), although some are single-line.
- Are declarative in nature.
- Cannot be nested.

A command is a keyword not preceded by the @ character. Commands are

- Procedural, describing the method and order of installation (creating directories, copying files, etc.).
- Usually grouped within an @FileSet...@EndFileSet descriptor block (although they can also exist independently).
- Executed in order, according to the script file.

Script files can be natural-language-enabled by using text within double quotes (in-line text). Such text is designated for language translation, including text that will be displayed in menu items, help text, etc.

Syntax Conventions

- Commands and arguments are not case-sensitive, but prompt strings are.
- A script parser recognizes the following special characters:

  = { } ( ) , : - ; > < ! [ ]

  If a special character is used within a name, the entire name must be surrounded by double or single quotation marks.

- ANSI escape sequences are recognized by the parser within double or single quotes. For example, '\\', "\n", and '\t' represent a backslash, a new line, and a tab, respectively.
- A NULL character (ASCII value 0) is not allowed anywhere in the file.

Comments

Syntax

```
; <text>
```

Description

A comment (semicolon followed by text) exists for documentation purposes only—it does not invoke a script file operation. Comment characters through the end of a line are read and discarded.

Example

```
; THIS IS A COMMENT
```

Descriptors

@CDName

Syntax

```
@CDName <name>
```

Description

@CDName specifies the name of the CD-ROM volume being installed. This name is passed to CDROM.NLM in a MOUNT command (CD MOUNT <*name*>) to mount the CD-ROM as a NetWare volume. This descriptor should only be used in a script on a CD-ROM. It should not be used in a diskette drive installation script. Copy CDROM.NLM to the STARTUP directory as specified in the BOOT fileset.

Example

```
@CDName NETWARE5
```

The installation will use the string NETWARE5 when trying to mount the CD-ROM as a NetWare volume. After successfully loading CDROM.NLM, it will issue CD MOUNT NETWARE5 at the system console.

Syntax

```
@DeletePartition <comma-separated list of partition types>
```

Description

@DeletePartition deletes the existing partitions that you specify. In contrast, the installation NLM's automatic partition creation deletes (by default) existing partitions of the first two types listed below. In both cases, all active partitions are left untouched.

Partition Type	Description	Deleted?
64	NetWare 286	Yes
65	NetWare 386	Yes
01	DOS, 12-bit FAT	No
04	DOS, 16-bit FAT	No
05	Extended partition	No
06	DOS Huge partition	No

If any of these partitions should not be deleted, or if additional partitions need to be added, use @DeletePartition to replace the list.

Example

If the following appears in NETMAIN.ILS, only NetWare 386 partitions are eligible for deletion:

```
@DeletePartition 65
```

Syntax

```
@Driver
  Help: "Help description for finding drivers"
  [Dir: <DSK | LAN>, <subdirectory name>, <vol label>,
  <disk description>]
  [Dir: <DSK | LAN>, <subdirectory name>, <vol label>,
  <disk description>]
  ...
@EndDriver
```

Description

@Driver specifies help for finding drivers in maintenance-mode installations, or finding disks and directories that contains drivers. Use @Driver in the NETMAIN.ICS file.

Variable	Description
<*subdirectory name*>	Specifies the directory to read. For floppy or network installations, the subdirectory is typically NULL. For a CD-ROM, the subdirectory is usually the common directory where drivers are found.
<*vol label*>	Specifies the name of the corresponding media volume.
<*disk description*>	Specifies the printed name on the media that contains the include file.

Example

```
@Driver
  Help: "If installing from a floppy, disk drivers will be
  on disk NetWare-2 and LAN drivers will be on disk
  NetWare-3. If installing from CDROM, all drivers will be
  in the directory <cdrom drive>:\NW40\SERVER\DRIVERS."
  Dir: DDSK, '', 'NETWARE-2', 'NetWare-2'
  Dir: LAN, '', 'NETWARE-3', 'NetWare-3'
@EndDriver
```

Disk NetWare-2 will be scanned for .DSK and .DDI files; disk NetWare-3 will be scanned for .LAN and .LDI files.

@FileSet
@EndFileSet

Syntax

```
@FileSet
  Description: "...[fileset description]..."
  Class: {MANDATORY | CORE | OPTIONAL | OPTIONAL_OFF |
  BOOT}
  Name: <text name - no spaces allowed>
  Bytes: <total transfer size for this fileset>
  [DiskBytes: <total disk space for this fileset>]
  [Attribute: <attribute1>, <attribute2>]
  [Flags: <flags>]
  <fileset commands...See "Syntax" on page 91.>
@EndFileSet
```

Description

@FileSet specifies a logically similar group of files to be copied and the necessary commands for copying and installing those files.

The Description and DiskBytes information for OPTIONAL and OPTIONAL_OFF classes of filesets is displayed in a menu before copying. The description for all filesets are displayed during the file copy. Bytes are used to update a status bar representing the total bytes to copy for all selected filesets, which is used in displaying the status bar.

Class specifies how the fileset will be used:

File	Action
MANDATORY	Always executed.
CORE	Always copied in a sequential install, but optionally copied in the selective install/maintain mode.
OPTIONAL	Displayed to the user for selection. They are defaulted to on (selected for copy).

File	Action
OPTIONAL_OFF	Displayed for the user to choose, but are defaulted to off.
BOOT	Copied, depending on the kind of installation and whether the server was launched from the directory that the user is installing to.

For all classes, as the files are copied, the file group status is also displayed, including the file being copied and the percent of all the filesets that have been completed.

Name is used to relate logical filesets together, even when they are declared in different included files. If multiple filesets exist with the same logical name, there can be only one description and one class for all of them. For example, if multiple filesets with the name OS2 are declared, and the class is OPTIONAL, only one description entry for the OS2 files is presented for the user to select. If the user selects it, all filesets with the name OS2 are copied.

BOOT files are files such as SERVER.EXE, SERVER.MSG, NWCONFIG.NLM, NWCONFIG.MSG, NUT.NLM, and NUT.MSG. The BOOT filesets are for internal use only: product files should not have any BOOT fileset declarations. Boot files cannot be compressed or renamed.

Attribute value indicates the final file attributes.The default value for attribute1 is 00060081 (Read Only, Shareable, Rename Inhibit, Delete Inhibit).

For a DOS file:

Attribute1	Meaning
00000001	Read Only
00000002	Hidden
00000004	System
00000008	Volume Label

For a NetWare file:

Attribute1	Meaning
00000001	Read Only
00000002	Hidden
00000004	System
00000008	Execute
00000010	Subdirectory
00000020	Archive
00000080	Shareable
00000700	Smode
00001000	Transaction
00004000	Read Audit
00008000	Write Audit
00010000	Immediate Purge
00020000	Rename Inhibit
00040000	Delete Inhibit
00080000	Copy Inhibit

The meaning for Attribute2 is 00000001 (do not decompress or process file). The default for Attribute2 is zero, which should be used in all usual circumstances.

Flag	Meaning
00000001	Do only for a custom install.
00000002	Do in first-phase copy (bit reset = second-phase).
00000004	Do only if NetWare for OS/2*.

Flag	Meaning
00000008	Do only if native (not SFT III™ or NetWare for OS/2).
00000010	Do only if SFT III.
00000020	Do only for a simplified installation.
00000040	Do only for maintenance mode.
00000080	Do only for upgrade.
00000100	Do only if installing from front end (NWNSTLL.EXE).
00000200	Do only if not installing from front end (NWNSTLL.EXE).

Example

```
@FileSet
  Class: CORE
  Name: SYSTEM_FILES
  Description: "System Files"
  Bytes: 10192463
  CopyToServer 'SYSTEM', '*.*', 'SYSTEM-1', "NetWare
  System-1 Files"
@EndFileSet
```

In this example, *.* files are copied from volume SYSTEM-1:, and the percent of complete status for this fileset is displayed during copying.

@IncludeFile
@EndIncludeFile

Syntax

```
@IncludeFile
  File: <filename>
  [DiskDescription: "...[disk description]..."]
  [DiskVolume: 11 char vol name>]
@EndIncludeFile
```

Description

@IncludeFile specifies a file (<*filename*>) with additional commands to parse and execute. DiskDescription specifies the printed name on the media that shows where the include file is located; DiskVolume is the corresponding media volume name.

The command file's directory is searched first. If the include file is not there, the specified disk volume is checked. If <*filename*> is a <*path*>\<*file*> combination, the command file directory is checked first for <*file*>, then the disk volume at the <*path*> offset is checked. If <*file*> includes wildcard characters, all files matching the pattern are included. If the filename has wildcards but no files match, none are included, and no error results. If the filename does not have wildcards, and the file is not present, an error occurs. If the filename is in single quotes and contains %{NWLANG}, %{NWLANG} is replaced by the language number (4=English).

Example

For the NETMAIN.ILS file, where NETMAIN.ICS is not in the same directory as NETMAIN.ILS:

```
@IncludeFile
  File: NETMAIN.ICS
  DiskDescription: '[1]'
  DiskVolume: SYSTEM-1
@EndIncludeFile
```

The file is parsed just after the @EndIncludeFile statement. If the installation cannot find NETMAIN.ICS in the directory where NETMAIN.ILS is, it prompts for the CD, then reads and parses NETMAIN.ICS from the CD.

@MessageFile
@EndMessageFile

Syntax

```
@MessageFile
 File: <filename>
 [Version: <message file version number>]
@EndMessageFile
```

Description

@MessageFile specifies a file, usually with the extension .MSG, containing message strings. It is typically located in the same directory as the script file or in an NLS subdirectory. When <*message file version number*> name variables are encountered when the remainder of the script file is parsed (including @Include files), the symbols are substituted with definitions from the message file.

The search order of message files is:

1. The current directory.
2. The NLS directory corresponding to the current language.
3. The default definitions of <*message file version number*> "string."

The Version number is eventually used to synchronize the script file with the messages. If no version is specified, version checking is not performed. Currently, no version checking is performed, regardless of whether the @Version descriptor is present.

Example

Using the NETMAIN.ILS file (in the same directory as NETMAIN.MSG):

```
@MessageFile
  File: NETMAIN.MSG
@EndMessageFile
```

NETMAIN.MSG is read into memory and used to dereference <*message file version number*> strings.

Syntax

```
@StartAppObject
  AppPlatform: <Supported Platforms>
  AppName: <Name of the object to create>
  AppPath: <UNC Path to the executable>
  [AppDescription: "...Text..."]
  [AppContact: <List of contacts>]
  [AppMapping: <List of drive mappings>]
  [AppFlags: <>]
  [AppIcon: <Filename of Icon>]
  [AppCaption: "...Text..."]
  [AppParameters: <List of Parameters for AppPath>]
  [AppPrinter: <Printer Ports>]
  [AppShutdownScript: <Filename of Script>]
  [AppStartupScript: <Filename of Script>]
  [AppWorkingDir: <Startup directory for App>]
  [AppUser: <List of users>]
  [AppAdminNotes: "...Text..."]
  [AppFaultTolerance: <List of fallover apps>]
  [AppLoadBalancing: <List of load balancing apps>]
@EndAppObject
```

Description

Use @StartAppObject to create NDS objects for use with the Novell® Application Launcher™ (NAL) product. It is used with NetWare for Small Business.

Variable	Description
AppPlatform	The list of supported platforms the Application object can run on.
AppName	The NDS name of the object as viewed in NetWare Administrator.
AppPath	The UNC path and filename of the executable to be launched by the object.
AppDescription	The description that shows on Novell Application Launcher.

Variable	Description
AppContact	The list of usernames to contact for resolving issues with the Application object.
AppMapping	A list of drive mappings that are made for the application to run correctly.
AppIcon	The filename and location of the applications icon.
AppWorkingDir	The startup directory for the application.
AppUser	A list of users authorized to run the application.

Example

```
@StartAppObject
  AppPlatform: 3x 95 NT
  AppName: "Netscape Install"
  AppPath: "SYS\\PUBLIC\\NETSCAPE\\NTSCINST.EXE"
  AppCaption: "Netscape Install"
  AppDescription: "Install the Netscape Internet browser"
  AppContacts: admin
  AppFlags: "No Cleanup"
  AppIcon: "sys:public\\netscape\\netscape.ico"
  AppWorkingDir: "sys\\public\\netscape"
  AppUsers: "All Users" Action Standard Desktop
@EndAppObject
```

This example creates an NDS Application object named Netscape* Install. When NAL is run, this object will let the user install it to their desktop.

@Other
@EndOther

Syntax

```
@Other
  Description: "...[text]..."
  [DiskDescription: "...[disk description]..."]
  [DiskVolume: <11 char vol name>]
  [File: <filename>]
  [Flags: <flags>]
@EndOther
```

Description

Description is the menu description displayed for the @Other configuration option. This label must be first in the set of labels for a particular @Other option.

File specifies the *.IPS file to execute, or the NLM to load to install the @Other option.

DiskDescription is the prompt name that will be displayed to the user to prompt for another diskette.

DiskVolume is the volume name for the disk specified in DiskDescription. If the DiskVolume label is present, the user is prompted for the specific volume using DiskDescription from drive A: (the user can change the drive). This continues until the file is found or the user cancels. If no DiskVolume is present, Install prompts for another diskette in drive A: and continues until the file is found (if File is present) or the user cancels. If File is not specified, installation looks for any *.IPS or PINSTALL.NLM files.

Flag	Meaning
00000001	Load product environment (clib, btrieve, product.dat).
00000002	Product included in product box (NETMAIN.ILS description).
00000004	Display product should be displayed in simplified installation.
00000008	Display product only in simplified installation.
00000010	Display product only in maintenance installation.
00000020	Display product only in upgrade.
00000040	Display product only in NetWare for OS/2 install.
00000080	Display product only in native installation.
00000100	Display product only in SFT III installation.
00000200	Display product only if started from front end (NWNSTLL.EXE).
00000400	Display product only if not started from front end (NWNSTLL.EXE).

If the other filename extension is .IPS, the file is assumed to be a script and is interpreted. If the other filename extension is .NLM, the file is loaded and executed as an NLM.

Example

```
@Other
  Description: "Install NetWare for Macintosh"
  DiskDescription: 'NWM System 1'
  DiskVolume: 'MAC-1'
  File: PINSTALL.NLM

  Description: "Install TCP/IP Protocol"
  DiskDescription: 'TCPIP System 2'
  DiskVolume: 'TCPIP-2'
  File: TCPIP.IPS
@EndOther
```

The Install NetWare for Macintosh* and Install TCP/IP Protocol menu items are displayed. If NetWare for Macintosh is selected, the user is prompted for the NWM System 1 CD, then PINSTALL.NLM on volume MAC-1: is executed. If TCP/IP is selected, the user is prompted for the TCPIP System 2 CD, then the command file TCPIP.IPS is interpreted.

@SyntaxVersion

Syntax

`@SyntaxVersion <major version>.<minor version>`

Description

The SyntaxVersion string represents the version of the script file parser. The SyntaxVersion string should be present for all script files, and the syntax version of the parser must match that of the file.

Example

```
@SyntaxVersion 1.02
```

The internal syntax string in the installation utility will be compared with this string. If they do not match, the installation will prompt for a different file. All included scripts with a @SyntaxVersion descriptor will be checked for matching version strings (see "@IncludeFile @EndIncludeFile" on page 97).

@StartSchemaMod
@EndStartSchemaMod

Syntax

```
@StartSchemaMod
 SchemaFileName: <Filename>, <Description>
@EndSchemaMod
```

Description

@StartSchemaMod lets NDS schemas be extended through installation scripts.

SchemaFileName specifies the filename for the schema file, along with a Description for displaying to the screen during the modification.

Example

```
@StartSchemaMod
  SchemaFileName: "sys:\\system\\schema\\nwadmin.sch",
  "NetWare Administrator"
@EndSchemaMod
```

In this example, the NWADMIN.SCH file extends the NDS schema and displays "NetWare Administrator" on the screen during the installation of Directory Services.

@TimeZone
@EndTimeZone

Syntax

```
@TimeZone
  Description: <tz description>
  Standard: <tz abb.>, <tz adj.>
  [Daylight: <dst abb.>, <dst adj.>]
  [DSTStart: <start rule>, <seq number>, <day number>,
  <weekday number>, <month number>, <time>]
  [DSTEnd: <end rule>, <seq number>, <day number>,
  <weekday number>, <month number>, <time>]
@EndTimeZone
```

Description

Description specifies a time zone description to be displayed to the user during time services installation.

Standard specifies information to be displayed to the user once the time zone has been selected. This includes the time zone abbreviation and the adjust +/-x:xx:xx from GMT (Greenwich Mean Time) or UTC (Universal Coordinated Time).

If Daylight exists, it specifies that daylight savings time exists for part of the year. It also specifies the daylight abbreviation, and the adjust +/-x:xx:xx from standard time.

DSTStart and DSTEnd specify the rule (w for weekday of month or d for day of month), the sequence number (1-relative), day number (1-relative), weekday number (1-relative), month number (1-relative), and time on a 24-hour clock for daylight savings time.

Sequence number means:		
1	First	(ignore day number)
2	Second	(ignore day number)
3	Third	(ignore day number)
4	Fourth	(ignore day number)
5	Fifth	(ignore day number)
6	Last	(ignore day number)
7	>= day number	
8	<= day number	

Example

```
@TimeZone
  Description: "United States, Mountain Time Zone",
  Standard: "MST", '7:00'
  Daylight: "MDT", '+1:00'
  DSTStart: w, 1, 1, 1, 4, '2:00'
  DSTEnd: w, 6, 1, 1, 10, '2:00'

  Description: "United States, Central Time Zone",
  Standard: "CST", '6:00'
  Daylight: "CDT", '+1:00'
  DSTStart: w, 1, 1, 1, 4, '2:00'
  DSTEnd: w, 6, 1, 1, 10, '2:00'
@EndTimeZone
```

Two time zone descriptions are displayed. When one is selected, the time parameters are filled in according to the above information, and the user can verify or edit them.

@TranslateModule
@EndTranslateModule

Syntax

```
@TranslateModule
 <old driver name1>,<new driver name1>,
 <old driver name2>,<new driver name2>,
 ...
@EndTrandlateModuel
```

Description

@TranslateModule is used internally by NETMAIN.ILS. It specifies that a disk or LAN driver name has been changed for the previous release. The name will be changed in the STARTUP.NCF or AUTOEXEC.NCF file during an upgrade.

@Version

Syntax

```
@Version <major version>.<minor version>
```

Description

The @Version string represents the version of the script file (and its corresponding script data). For the NETMAIN.ILS script file, @Version must be greater than or equal to the expected version of the installation NLM. If any included script files contain version numbers, all the numbers must match. @Version should be incremented each time a new script is released.

In product scripts, the version string is entirely optional. A version string in a product script is not compared with the installation NLM, but all included product scripts must have matching version strings.

Example

```
@Version 1.01
```

The version string in the installation is compared with this version string. If they do not match, the installation prompts for a different file. All included scripts with a @Version descriptor will be checked for matching version strings (see "@IncludeFile @EndIncludeFile" on page 97).

Commands

Command

Syntax

```
Command <command handler name>
```

Description

Command specifies the name of an optional command handler for commands not included in this file. The command handler name must be eight characters or less and correspond to the name of an NLM that exports external command entry points.

Example

```
Command ICMD.NLM
```

ICMD.NLM provides external command support (see “External Command Syntax” on page 114). It supports the command syntax described in this document, plus additional commands.

CopyDriverToServer

Syntax

```
CopyDriverToServer <target_directory_path>, <file_spec>,
<source_volume>, <source_description>
```

Description

CopyDriverToServer replaces driver files in <*target_directory_path*> with new driver files. The source files are assumed to be those on the distribution CD (which might have been copied to a user-specified directory).

<*file_spec*> is the name of the file to be copied, and can be of the form *.LAN, *.DSK, <*directory*>/*.LAN or <*directory*>/*.DSK. <*source_volume*> specifies the name of a diskette volume label and <*source_description*> specifies the string used to prompt for the diskettes. If the installation source is a CD-ROM or network drive, <*source_volume*> and <*source_description*> are ignored (although these fields must be present in the command).

Example

```
CopyDriverToServer SYSTEM, '*.LAN', 'NETWARE-3', “NetWare
  diskette [3]”
```

The *.LAN files are copied to the SYS:SYSTEM directory.

CopyDriverToStartup

Syntax

```
CopyDriverToStartup <file_spec>, <source_volume>,
<source_description>
```

Description

CopyDriverToStartup replaces driver files in the startup or boot directory (where STARTUP.NCF is) with new driver files. The source files are assumed to be those on the distribution CD (which might have been copied to a user-specified directory). <*file_spec*> names the files to be copied, and can be *.LAN, *.DSK, <*directory*>/*.LAN or <*directory*>/*.DSK. <*source_volume*> specifies the name of a diskette volume label and <*source_description*> specifies the string used to prompt for the diskette. If the installation source is a CD or network drive, <*source_volume*> and <*source_description*> will be ignored (although these fields must be present in the command).

Example

```
CopyDriverToStartup '*.LAN', 'NETWARE-3', "NetWare
  diskette [3]"
```

The *.LAN files are copied to the user-designated DOS startup directory where the server boot files reside.

CopyToServer

Syntax

```
CopyToServer <target_directory_path>, <file_spec>,
<source_volume>, <source_description>
```

Description

CopyToServer specifies a file (or a group of files using wildcards) to copy to the system volume of the server.

<*target_directory_path*> is a directory path relative to SYS: where the files will be copied. <*file_spec*> names the files to be copied; it might be in the "directory/file" form to copy files from a subdirectory. <*source_volume*> specifies the name of a diskette volume label and <*source_description*> specifies the string used to prompt for the diskettes. If the installation source is a CD-ROM or network drive, the <*source_volume*> and <*source_description*> fields are ignored (although these fields must be present in the command).

Example

```
CopyToServer 'SYSTEM', '*.*', 'SYSTEM-2', "NetWare
  diskette [4]"
```

Files matching the DOS file specification (including DOS wildcard names) are copied from the volume SYSTEM-2 (or from a user-specified hard disk or network directory) to directory SYS:SYSTEM. If the volume SYSTEM-2 is not present, the user will be prompted to insert the NetWare CD.

CopyToStartup

Syntax

```
CopyToStartup <file_spec>, <source_volume>,
<source_description>
```

Description

CopyToStartup copies files to the startup or boot directory (where STARTUP.NCF is). The source files are assumed to be those on the distribution diskettes (which might have been copied to a user-specified directory).

<*file_spec*> names the files to be copied. To copy files from a subdirectory, <*file_spec*> can be of the form "directory/file". <*source_volume*> specifies the name of a diskette volume label and <*source_description*> specifies the string used to prompt for the diskettes. If the installation source is a CD or network drive, <*source_volume*> and <*source_description*> are ignored (although these fields must be present in the command).

Example

```
CopyToStartup 'SERVER.EXE', 'SYSTEM-3', "NetWare diskette
  [3]"
```

SERVER.EXE is copied to the user-designated DOS startup directory where the server boot files will reside.

Exec

Syntax

Exec <*filename*>

Description

Exec issues the operating system command LOAD <*filename*>, pauses until the child NLM unloads, then continues executing. To load from other subdirectories of the system volume besides \SYSTEM, specify a filename (SYS:/<*directory*>/<*file*> and <*directory*>/<*file*>).

Example

```
Exec MY.NLM
```

MY.NLM will be loaded and executed. Exec will follow the normal NLM search path to find MY.NLM. Control will return to the installation after Exec exits and unloads.

ProductRecord

Syntax

ProductRecord <*product ID (string)*>, <*record type*>, <*record data*>

Description

ProductRecord creates a record in the product database with the ID string <*product ID*> and the record type and data indicated.

Record Type	Data
0	Version string (10 char max)
1	Product description (60 char max)

Record Type	Data
[2]	Configuration NLM command line(255 char max)
[3]	Configuration text file specification (255 char max)
[4]	Un-install NLM command line (255 char max)
[5]	Delete AUTOEXEC line (255 char max)
[6]	Delete STARTUP line (255 char max)
[7]	Delete file line (255 char max)

Example

```
ProductRecord MYPROD, 1, "Description for my product"
```

This creates a product description record for MYPROD with the quoted description.

RegisterLanguage

Syntax

```
RegisterLanguage <language ID>, <language name>
```

Description

RegisterLanguage renames <*language ID*> to the respective name <*language name*>. This exists typically to allow non-standard languages to be registered when they are installed.

Example

```
RegisterLanguage 17, TURKISH
```

This command aliases TURKISH to ID 17.

SaveLanguageFile

Syntax

```
SaveLanguageFile <file>
```

Description

SaveLanguageFile copies <*file*> in the DOS startup directory to the directory indicated by the old language ID <*startup directory*>/<*old language ID*>. This might involve creating a new directory. If the directory path for the new file does not exist, it will be created.

Example

```
SaveLanguageFile SERVER.MSG
```

If the current language ID is 4, SERVER.MSG in the DOS startup directory (BOOT directory) is copied to the subdirectory 4 in the DOS startup directory.

Spawn

Syntax

```
Spawn <file name>
```

Description

Spawn issues the operating system command LOAD <*file_name*>, then the installation continues to execute. See “Exec” on page 111.

Example

```
Spawn MY.NLM
```

The MY.NLM file is loaded and executed. Spawn follows the normal NLM search path to find MY.NLM. The installation and the child NLM both execute, sharing the CPU through context switches.

External Commands

This section describes commands provided by ICMD.NLM, a companion to the NetWare Installation NLMs (INSTALL.NLM and NWCONFIG.NLM). See “NetWare Installation Command Version Availability” on page 149 for a table showing the ICMD versions that support these commands.

Version 2.10 External Command Compatibility

If you intend to use ICMD.NLM 2.18 or greater, do not copy this new version to SYS:SYSTEM. Instead, copy it to a different directory and load it from there.

For example, the script commands below will copy ICMD.NLM to a temporary directory and load from there.

```
CopyToServer SYSTEM\TMP, ICMD.NLM, "", ""
  Command TMP\ICMD
```

Using this method ensures that the ICMD.NLM version in SYS:SYSTEM is the actual 4.10 version and that all built-in products that use it will work properly. The new ICMD.NLM might still be fully backward-compatible, but it has not yet been fully tested in that regard.

External Command Syntax

To use any of the external commands made available by the ICMD.NLM command handler, include the statement command [<*filepath*>] ICMD in the script file. Use CopyToServer to copy ICMD.NLM to a temporary directory (for example, SYS:SYSTEM\TEMP), then use command to run it from there (for example, command SYS:SYSTEM\TEMP\ICMD). This removes any compatibility problems associated with using the standard ICMD.NLM in SYS:SYSTEM. You can create then add additional commands to your own version of ICMD.NLM (after you rename the file) to perform application-specific functions.

Commands

Activate

Syntax

```
Activate
```

Description

Activate causes the installation screen to be activated.

Example

```
Activate
```

The installation screen will be displayed for further script processing.

AppendFile

Syntax

```
AppendFile <src var name>, <src file spec>, <src vol name>,
<src descr>, <dest var name>, <dest dir>, <dest vol name>,
<dest descr>
```

Description

AppendFile appends the contents of the source file onto the destination file. It creates a temporary file that is as large as the destination file, then renames that file to the destination file. Wildcard characters for <*src file spec*> are not allowed.

Example

```
AppendFile input, file1, 'NETWARE-1', "NW [1]", output,
  file2, '', ''
```

The user is prompted for diskette NW [1], which is verified by checking for volume label NETWARE-1. The FILE1 on that diskette is appended to FILE2 in the directory corresponding to output.

CheckFile

Syntax

```
CheckFile <var name>, <file spec>, <vol name>, <description>
```

Description

CheckFile checks for the existence of <*file spec*> at a path determined by the value of <*var name*>. If the path is a floppy, and <*vol name*> is non-null (not ''), the user will be prompted for disk <*description*> if it is not already in the drive.

If the file is found, the value of variable NWSTATUS will be zero; otherwise it will be non-zero.

Example

```
CheckFile NWSRC, file1.dat, '', ''
GotoIfNEqual '%{NWSTATUS}', 0, FileNotFound
File found
.
.
Label FileNotFound
```

In this example, a check is made for a file (FILE1.DAT). If the file is found at the path determined by NWSRC, execution continues on the next line. If the file is not found, control branches to label FileNotFound.

Config

Syntax

```
Config <flag>
```

Description

Config creates a configuration file with available default information from the installation. If <*flag*> is 1, AUTOEXEC.NCF is created on SYS:SYSTEM; if 0, STARTUP.NCF is created in the server boot path.

Example

```
Config 1
```

This command creates AUTOEXEC.NCF, with server name, internal net address, configured LAN driver load and bind commands, directory service and time services information. The existing AUTOEXEC.NCF on SYS:SYSTEM is overwritten.

Console

Syntax

```
Console <text>, <delay in seconds>
```

Description

Console enters the text on the system console screen and delays <*delay in seconds*> before returning control back to the calling process. Because the system installation process executes in a different thread from the console, it is possible that the next command might finish before this Console command does.

Example

```
Console 'load ne2000 frame=ethernet_802.3 int=3 port=300',
  10
Console 'bind ipx to ne2000 net=1240000', 3
```

These commands load and bind an NE2000™ driver to IPX™.

CopyFile

Syntax

```
CopyFile <recurse flag>, <directory flag>, <empty flag>,
<attribute flag>, <attributes>, <src var name>, <src file
spec>, <src vol name>, <src descr>, <dest var name>, <dest
dir>, <dest vol name>, <dest descr>, [<conditional copy flag>]
```

Description

CopyFile copies files (wildcards are allowed in <*src file spec*>) from the source path to the destination path.

<*recurse flag*> (0|1)

Value	Description
0	Do not copy subdirectories recursively.
1	Copy subdirectories recursively.

<*directory flag*> (0|1|2|3)

Value	Description
0	Source is a file.
1	Source (including <*src file spec*>) is a directory; copy directory contents to destination.
2	Source and the destination are files (file-to-file copy). <*recurse flag*> must be 0 if <*directory flag*> is 2
3	Source (including <*src file spec*>) is a directory; copy directory contents to destination only if that directory already exists on the destination.

<*empty flag*> (0|1)

Value	Description
0	No error if no files are found.
1	Error if no files are found.

<*attribute flag*> (0|1|2)

Value	Description
0	Use the default file set attributes.
1	Use the source file's attributes.
2	Use the attribute specified by <attributes> (must be used to copy to DOS drive).

<attributes>

For a DOS file:

Attribute	Meaning
00000001	Read Only
00000002	Hidden
00000004	System
00000008	Volume Label

For a NetWare file:

Attribute	Meaning
00000001	Read Only
00000002	Hidden
00000004	System
00000008	Execute
00000010	Subdirectory
00000020	Archive
00000080	Shareable
00000700	Smode
00001000	Transaction
00004000	Read Audit
00008000	Write Audit
00010000	Immediate Purge
00020000	Rename Inhibit
00040000	Delete Inhibit
00080000	Copy Inhibit

Reserved path names (<*src*|*dest var name*>) are NWSRC, NWDST and NWBOOT, the standard source, destination and boot path for NetWare files. <*dest dir name*> can be a null string.

<*conditional copy flag*>

Value	Description
0 (or not present)	Unconditionally copy this file.
1	Copy if destination file is not present.
2	Copy if destination file is not present; if it is, prompt user (no version check).
3	Copy if destination file is not present or if the source file version is newer than the destination file version.
4	Copy if destination file is not present or if the source file version is newer than the destination file version; otherwise, prompt the user.
5	Copy if destination file is present (no version check).
6	Copy if destination file is present and source file version is newer than destination file version.
7	Copy if destination file is present and source file version is newer than destination file version; otherwise, prompt the user.
8	Copy if destination file is not present or if source file version is newer than destination file version (refresh with new files).
9	Copy only if destination file is present and source file version is newer than destination file version (refresh existing files).

Example

```
CopyFile 0, 0, 1, 0, 0, input, *.*, 'NETWARE-1', "NW [1]",
  output, '', '', ''
```

This is a continuation of the examples for "GetPath" on page 129. The user is prompted for diskette NW [1], and the diskette is verified by checking for volume label NETWARE-1. All files on the diskette are copied to the subdirectory the user indicated for output.

CopyFloppyImage

Syntax

CopyFloppyImage <*floppy type flag*>, <*diskette name*>, <*src var name*>, <*src file spec*>, <*src vol name*>, <*src descr*>, <*dest var name*>, <*dest dir*>, <*dest vol name*>, <*dest descr*>

Description

CopyFloppyImage copies a diskette image file sector by sector to the specified drive (destination path which must be a diskette drive).

Variable	Description
<*floppy type flag*> (0|1)	0 indicates a 3.5-inch high-density drive. 1 indicates a 5.25-inch high-density drive.
<*diskette name*>	The external label for the diskette.

Example

```
GetPath, dst, 2, 'A:', ''
CopyFloppyImage 0, "Disk 1", 3.5\WSDOS_1.IMG, '', '', dst,
  '', '', ''
```

The floppy image specified by the source path is copied to the diskette in the specified drive.

Delay

Syntax

```
Delay <delay in seconds>
```

Description

Delay relinquishes control for <*delay in seconds*> seconds before returning control back to the calling process. If a negative value is used, the delay is a random value from 0 to the absolute value of <*delay in seconds*>.

Example

```
Delay '-10'
```

This delays randomly from 0 to 10 seconds.

DiskReset

Syntax

```
DiskReset <var name>
```

Description

DiskReset tells DOS to rescan the directories on the <*var name*>floppy drive. This is necessary because the drive change interrupt is disabled while NetWare has control.

Example

Assuming myDir corresponds with A:\

```
DiskReset myDir
```

Drive A:\ will be rescanned.

Display_Text_File

Syntax

```
Display_Text_File <Path>, <Filename>, <Completion Code>
```

Description

Display_Text_File displays an ASCII text file. The user can scroll the text in any direction to read the entire contents of the file. Scroll bars automatically display if needed. The maximum file size is 30 KB for the file.

Example

```
GetPath sys, 1, 'SYS:', ''
Display_Text_File sys, "license.txt", ccode
```

The completion code will return a 0 (zero) for a successful return. Anything else is an error.

Display

Syntax

```
Display <0|1|2>, "...<text>..."
```

Description

Display displays a message, accompanied by a beep if the first argument is 1. If the argument is 2, the message is displayed as a copy status.

Example

```
Display 0, "Product XYZ installation is complete."
```

The text is displayed (without a beep), and the user can press Enter to continue.

Syntax

```
DSInst <time zone name>, <time server type>, <daylight
   flag>, <daylight start>, <daylight end>, <daylight
   offset in seconds>, <tree name>, <tree address>,
   <container object name>, <admin name>, <admin password>,
   <replica flag>, <upgrade bindery flag>, <root server
   flag>
```

Description

DSInst installs the directory with default time synchronization.

Variable	Description
<*time zone name*>	Time zone name (works as a system SET parameter)
<*time server type*>	Time server type (works as a system SET parameter): single, reference, primary, secondary
<*daylight flag*>	1: Daylight savings time exists in this time zone 0: Daylight savings time doesn't exist in this time zone
<*daylight start*>	“set parameter” daylight savings time start string. For Example First Sunday of April at 2:00:00 a.m
<*daylight end*>	“set parameter” daylight savings time end string. For Example Last Sunday of October at 2:00:00 a.m.
<*daylight offset in seconds*>	Typically this is 3600 (seconds per hour), or zero if daylight saving time is not applicable.
<*tree name*>	DS tree name
<*tree address*>	Internal IPX address of the SAP server for the tree
<*container object name*>	Full DN of container

Variable	Description
<*admin name*>	Full DN of Admin object (must have supervisor rights on the container)
<*admin password*>	Password for <*admin name*>
<*replica flag*>	1: Install a replica on this server; 0: don't install
<*upgrade bindery flag*>	1: Upgrade the bindery if it exists; 0: don't install
<*root server flag*>	1: Make this a new tree; 0: don't make this a new tree

Example

```
DSInstMST7MDT, SECONDARY, 1, '(APRIL SUNDAY FIRST 2:00:00
  AM)', '(OCTOBER SUNDAY LAST 2:00:00 AM)', 3600, URSA,
  FACE3234, 'O=Novell', 'CN=Admin.O=Novell', '', 1, 1, 0
```

Time services will be installed with this server as a secondary server using existing tree name URSA. Directory services will be installed, with this server in container O=Novell. This server will have a replica installed, and its bindery will be upgraded.

EraseFile

Syntax

```
EraseFile <directory flag>, <var name>, <file spec>, <vol
name>, <vol descr>
```

Description

EraseFile recursively erases the file or directories indicated in <*file spec*>. Wildcards are acceptable. In NetWare, files can be removed no matter what their attributes; in DOS, only normal files can be removed.

IMPORTANT: Use this command with care to delete only the intended data.

If <*directory flag*> is 1, <*file spec*> must be a directory name; the contents of the directory will be deleted, but the directory will not.

Example

If myDir corresponds to VOL1:\.

```
EraseFile 1, myDir, '', '', ''
```

The contents of directory VOL1:\ will be erased.

EditNCF

Syntax

EditNCF <*ncf path var*>, <*ncf file name*>, <*backup ext*>, <*ccode*>

Description

EditNCF edits the specified NCF file as follows:

- Writes any lines in the EditNCF_ADDLINE list at the beginning of the NCF file.
- Deletes any lines that match lines in the addline list.
- Comments out any lines containing keystrings in the EditNCF_KEYSTRINGS list.
- Any other lines are copied unchanged.

Example

```
GetPath startup, 1, 'C:\\NWSERVER', ''
EditNCF_KEYSTRING ''
EditNCF_KEYSTRING 'PK411.NLM'
EditNCF startup, 'startup.ncf', 'bak', ccode
```

The contents of C:\NWSERVER\STARTUP.NCF will be searched for a line containing PK411.NLM and will be commented out by the EditNCF command. Ccode will contain the following:

0 - Success
1 - Duplicate Extension
2 - Open Error
3 - Read Error
4 - Write Error
5 - NCF to Backup Extension Error
6 - Tmp to NCF Error

EditNCF_ADDLINE

Syntax

```
EditNCF_ADDLINE <LineToBeAdded>
```

Description

EditNCF_ADDLINE adds a line of text to the list of lines to be added at the top of the file being edited with EditNCF.

Example

```
EditNCF_ADDLINE ''
EditNCF_ADDLINE 'LOAD PK411.NLM'
EditNCF startup, 'startup.ncf', 'bak', ccode
```

The first line will clear the ADDLINE list. The second line will add the string "LOAD PK411.NLM" to the top of the STARTUP.NCF file.

EditNCF_KEYSTRINGS

Syntax

```
EditNCF_KEYSTRINGS <keystring>
```

Description

EditNCF_KEYSTRINGS defines a list of keystrings to be used by EditNCF when determining which lines to comment out. (A line will be commented out if it contains any of the keystrings).

NOTE: The list must be terminated with a null string (empty quotes) for the last entry. This command can be used multiple times. Each time replaces the previously defined list.

Example

```
EditNCF_KEYSTRINGS ''
EditNCF_KEYSTRINGS 'PK411.NLM', 'ADTRSTFX', 'AUTODUMP',
  'CLSSCRFX', 'EAPURGFX', 'EVNTRPFX', 'GETDIRFX', ''
EditNCF startup, 'startup.ncf', 'bak', ccode
```

The first line clears the KEYSTRING list. The second line adds the strings to the list to be commented out of the STARTUP.NCF file.

ExtractVersionNumbers

Syntax

```
ExtractVersionNumbers <version string>, <major var name>,
<minor var name>, <revision var name>
```

Description

ExtractVersionNumbers parses a version string and returns the major version number, minor version number, and revision number as integers. Version string is parsed and leading non-digits are ignored. The variables are filled with the major, minor, and revision values for the version string. If no revision value is in the version string, a zero is returned. If the revision value is a non-digit value, a numeric revision value is returned (i.e., A = 1, B = 2, etc.).

Example

```
SetVar version, "v2.11"
ExtractVersionNumbers version, major, minor, revision
```

The variable version contain the string "v2.11". ExtractVersionNumbers returns a value of 2 in the variable major, a value of 11 in the variable minor. And a value of 0 in the variable revision.

GetDOSServer

Syntax

```
GetDOSServer <var name>
```

Description

GetDOSServer returns the default DOS server number in <*var name*> on SFT III servers.

Example

```
GetDOSServerDOSServer
```

The value of the variable DOSServer will be either 0 or 1. If the value is anything else, the server is not running SFT III.

Syntax

```
GetPath <var name>, <path type>, <default>, <prompt>, [<base
var name>]
```

Description

GetPath prompts the user for a path using the prompt. The prompt string will be displayed, with "%s" (like C-language print formatting) being replaced by the <*default*> string.

The keystrokes to modify will be added to your prompt. Trailing backslashes are always removed before the path is displayed. <*var name*> is a variable name that can be used later in a CopyFile command. If <*prompt*> is a null string, the user is not prompted; the path variable is set. If <*base var name*> exists, it will be used as a basis for the new variable (the remote connection number will be maintained, etc.).

<*path type*>

Value	Description
1	Allow any path.
2	Allow floppy only (including remote).
3	Allow DOS only (including remote).
4	Allow NetWare local path only.
5	Allow local floppy only.

Example

```
GetPath input, 1, 'A:', "Files will be copied from %s\\."
```

The user will see the following prompts:

```
Files will be copied from A:\

Press <F3> to specify a different path.

Press <F4> to specify a remote path.

Press <Enter> to continue.
```

Goto

Syntax

```
Goto <label_name>
```

Description

Goto causes execution to continue at the first occurrence of label <*label_name*>.

The scope of a label is always local to a file set. Therefore, a Goto statement cannot transfer execution to a file set different from the current one.

Example

```
Goto Done
Label Done
```

The statement Goto Done will cause execution to continue at the label Done.

GotoIfEqual

Syntax

```
GotoIfEqual <larg1>, <arg2>, <label_name>
```

Description

GotoIfEqual causes execution to continue at the first occurrence of Label <*label_name*> if <*arg1*> is equal in value to <*arg2*>. Both <*arg1*> and <*arg2*> must be integers.

The scope of a label is always local to a file set. Therefore, a Goto statement cannot transfer execution to a file set different than the current one.

Example

```
GotoIfEqual 0, 1, Error
Goto Done
Label Error
```

Since 0 is not equal to 1, execution will not continue at label Error.

NOTE: GotoIfEqual performs an integer comparison, not a string comparison.

GotoIfNEqual
GotoIfGreater
GotoIfLess
GotoIfGrEqual
GotoIfLsEqual

Syntax

```
GotoIfNEqual <arg1>, <arg2>, <label_name>
GotoIfGreater <arg1>, <arg2>, <label_name>
GotoIfLess <arg1>, <arg2>, <label_name>
GotoIFGrEqual <arg1>, <arg2>, <label_name>
GotoIfLsEqual <arg1>, <arg2>, <label_name>
```

Description

These are identical to "GotoIfEqual" on page 130, except that execution branches only if (respectively):

<*arg1*> not equal to <*arg2*>
<*arg1*> is greater than <*arg2*>
<*arg1*> is less than <*arg2*>
<*arg1*> is greater than or equal to <*arg2*>
<*arg1*> is less than or equal to <*arg2*>

The scope of a label is always local to a file set. Therefore, a Goto statement cannot transfer execution to a file set different from the current one.

NOTE: GotoIf* performs an integer comparison, not a string comparison.

Is_NLM_Active

Syntax

```
Is_NLM_Active <NLM Name>, <Completion Code>
```

Description

Is_NLM_Active searches the loaded modules list and returns 1 if the NLM is loaded, or 0 if the NLM is not present in memory.

Example

```
Is_NLM_Active "Monitor.nlm", ccode
```

Label

Syntax

```
Label <label_name>
```

Description

Label serves only as a target of a goto statement with <*label_name*>.

IMPORTANT: The scope of a label is always local to a file set. Therefore, a Goto statement cannot transfer execution to a file set different from the current one.

Example

```
Goto Done
Label Done
```

The Goto Done statement will cause execution to continue at Label Done.

LangVar

Syntax

```
LangVar <default>, <lang var>, <src var dir name>, <src vol
name>, <src descr>
```

Description

LangVar sets a variable to the number of a language. It starts with <*src var dir name*>, then it looks for a subdirectory whose name is a number that matches the current server language number. If the subdirectory is found, <*lang var*> is set to that number; otherwise, <*lang var*> is set to <*default*>.

Example

Assuming the current language is 6:

```
LangVar 4, lang, NWSRC, '', ''
```

This command looks in the source install directory for a subdirectory named 6. Finding it, it sets lang to 6.

Menu

Syntax

```
Menu <var name>, <Prompt>, <# of menu options>, <default menu
option>, <first choice>, <first value>, <second option>,
<second value> [<third option>, <third value>,] [<fourth
option>, <fourth value>]
```

Description

Menu command will display a window with the <*Prompt*> showing in a header area. The <*choices*> will display in a scrolling area below the header. The <*# of menu options*> is 1 based. The <*default menu option*> is zero based. The entire command is terminated without a comma after the final value.

Example

Menu done, “Are you ready to continue?”, 2, 0,

```
“Yes”, 1,
“No”, 0
```

The Menu command displays a box on the screen with the Prompt “Are you ready to continue?” Below it the choices are “Yes” and “No.” The “Yes” option is highlighted as the default option. Upon selecting the option, the value will be placed in the variable. The values must be integers.

NLMExec
NLMExecIO

Syntax

```
NLMExec <block flag>, <command line>
```

Description

NLMExec executes the NLM specified by <*command line*>. If <*block flag*> is 1, NLMExec waits for the NLM to complete execution, then continues. NLMExec is the external command version of the “Exec” on page 111 command. NLMExecIO is identical to NLMExec except it executes the command line on the default DOS server on a NetWare SFT III server. See “SetDOSServer” on page 136 for more information.

Example

```
NLMExec 1, PINSTALL
```

This command executes PINSTALL.NLM and waits for it to finish.

Partitions

Syntax

```
Partitions <delete flag>
```

Description

Partitions creates NetWare partitions as needed on all available devices. If <*delete flag*> is 1, all existing nonbootable partitions will be deleted. If <*delete flag*> is 0, they will be left intact. The minimum allowable partition size is 1 MB, according to default rules (see "@DeletePartition" on page 92).

Example

For one existing device with a DOS (active) partition and a NetWare 286 partition:

```
Partitions 1
```

This deletes the 286 partition and creates a NetWare partition in the remaining space.

Quit

Syntax

```
Quit
```

Description

Quit will exit the script processing at this point. No further processing will take place.

Example

```
GotoIfEqual %{variable1}, %{true}, Continue_On
Quit
```

Label Continue_On

ReadProductRecord

Syntax

```
ReadProductRecord <productID>, <recordtype>,
<recorddatavarname>, <ccode>
```

Description

ReadProductRecord reads the record in the products database matching the <*productID*> and <*recordtype*> specified and returns the corresponding data. This command is the counterpart to the ProductRecord command. See “ProductRecord” on page 111 for more information on record types.

Example

```
ReadProductRecord MYPROD, 0, version, ccode
```

The command will search the products database and return the value for <*version*> for product 'MYPROD'.

Read_Var_File

Syntax

```
Read_Var_File <path>, <filename>, <completion code>
```

Description

Read_Var_File will read the contents of the filename and create variables for script processing. The file is an ASCII formatted file with the content as follows:

“Variable1=true”
“Variable2=false”
“Variable3=Servername”

Example

```
Read_Var_File “sys:system”, “varfile.txt”, ccode
```

NOTE: Ccode will return the file open status. 0 is success, anything else is an error. If the variable to the left contains any spaces, leading or trailing, the spaces will become part of the variable name and must be included with the variable when referencing it.

SetDir

Syntax

```
SetDir <var name>, <relative dir>, <vol name>, <vol descr>,
<attributes>, <0|1>, <rights>
```

Description

SetDir creates directories (if they do not already exist), according to the <*var name*> obtained from "GetPath" on page 129. The <*relative dir*> value can be null. <*attributes*> are the attributes to use when creating the directory; 0 indicates normal attributes. If the <*0|1*> argument is 1, the server container object (or everyone) is given <*rights*> rights to the directory.

Example

If myDir corresponds to VOL1:\

```
SetDir myDir, DOC, '', '', 10, 1, 0
```

The directory VOL1:\DOC will be created and all applicable users will have rights to it.

SetDOSServer

Syntax

```
SetDOSServer <engine number>, <ccode>
```

Description

SetDOSServer specifies the <*engine number*> to be made the default DOS server (valid numbers are 0 and 1). This will set the default server used by the OS APIs.

NOTE: DiskReset and GetPath can be directed to a particular IOEngine by first setting the DOS server as needed. This allows you to essentially map a path variable to a particular IOEngine. From that point on, that path variable will be associated with that IOEngine regardless of what the default DOS server is set to. Therefore, any commands that use path variables will target the IOEngine associated with the path variable rather than the default.

Example

```
SetVar IOEngineNumber, 0
SetDOSServer IOEngineNumber, ccode
```

The server's default IOEngine 0 will now be the default DOS server. Ccode will have a 0 for its value on success and a -1 for a failure.

SetLConfigLang

Syntax

```
SetLConfigLang <value>
```

Description

SetLConfigLang looks for the LCONFIG.SYS file in the server boot directory. The default server language ID number is changed to <*value*>.

Example

```
SetLConfigLang 9
```

The default server language ID number in the LCONFIG.SYS file is changed to 9 (Japanese).

SetVar

Syntax

```
SetVar <variable name>, <variable value>, [<Prompt>]
```

Description

SetVar sets a variable with <*variable name*> to the <*variable value*> string value. If the value does not exist, one is created. The variable value cannot exceed 127 characters in length.

Example

```
SetVar message, "Hello World!"
```

This causes a variable to be created with name "message," and its value set to the string "Hello World!"

NOTE: The Prompt option is only available in version 3.25+.

SGotoIfEqual
SGotoIfNEqual
SGotoIfGreater
SGotoIfLess
SGotoIfGrEqual
SGotoIfLsEqual

Syntax

```
SGotoIfEqual <arg1>, <arg2>, <label_name>
SGotoIfNEqual <arg1>, <arg2>, <label_name>
SGotoIfGreater <arg1>, <arg2>, <label_name>
SGotoIfLess <arg1>, <arg2>, <label_name>
SGotoIfGrEqual <arg1>, <arg2>, <label_name>
SGotoIfLsEqual <arg1>, <arg2>, <label_name>
```

Description

SGotoIfEqual causes execution to continue at the first occurrence of Label <*label_name*> if <*arg1*> is equal in value to <*arg2*>. Both <*arg1*> and <*arg2*> must be integers.

SGotoIfNEqual, SGotoIfGreater, SGotoIfLess, SGotoIfGrEqual, and SGotoIfLsEqual are identical to SGotoIfEqual except that execution branches only if (respectively):

<*arg1*> not equal to <*arg2*>
<*arg1*> is greater than <*arg2*>
<*arg1*> is less than <*arg2*>
<*arg1*> is greater than or equal to <*arg2*>
<*arg1*> is less than or equal to <*arg2*>

The scope of a label is always local to a file set. Therefore, an SGoto statement cannot transfer execution to a file set different from the current one.

NOTE: The SGotoIf* commands have the same options as the GotoIf* commands, but the SGotoIf* commands perform a string comparison instead of an integer comparison.

Example

```
SGotoIfEqual 0, 1, Error
Goto Done
Label Error
```

Since 0 is not equal to 1, execution will not continue at label Error.

SpaceCheck

Syntax

```
SpaceCheck <path var name>, <value>
```

Description

SpaceCheck looks at <*path var name*>. If it matches a NetWare volume, SpaceCheck checks whether the volume is large enough for <*value*> bytes, and whether the volume has enough free space for <*value*> bytes. If the volume is not large enough, the installation aborts. If there is not enough free space, the user is alerted and allowed to continue.

Example

Assuming srcVar corresponds to VOL1:\FILES

SpaceCheck srcVar, 1000000

Volume VOL1: will be checked for one million bytes of free space.

NOTE: ICMD.NLM versions 3.25 and older do not support SpaceCheck on DOS drives. Newer versions support this drive option.

SrchNCF

Syntax

```
SrchNCF <ncf path var>, <ncf file name>, <keystring>,
<foundline var name>
```

Description

SrchNCF searches an .NCF file for line containing a substring matching <*keystring*> (non case-sensitive). If found, the line of text will be returned; otherwise <*foundline var name*> will be empty.

NOTE: The same value as not found will be returned if an error is encountered (for example, invalid filename, read error, etc). It is assumed that this command will be used before calling "EditNCF" on page 126 which returns a more detailed completion code that can be acted on as necessary.

Example

```
GetPath sys, 1, 'SYS:\\', ''
SrchNCF sys, 'SYSTEM\\AUTOEXEC.NCF', "Load Monitor", ccode
```

The SrchNCF command will search the SYS:\SYSTEM\AUTOEXEC.NCF file for the string 'Load Monitor' and return the line number in ccode if found.

ValueSet
ValueAdd

Syntax

```
ValueSet <var name>, <value>
Value Add <var name>, <value>
```

Description

ValueSet sets, and ValueAdd adds to, the numeric contents of variable <*var name*>.

Example

```
ValueSet srcVar, 1
```

The variable srcVar is created if it did not exist and is set to 1.

Volumes

Syntax

```
Volumes <minimum vol SYS: size in sectors>
```

Description

Volumes creates and mounts volumes on this server, with the volume SYS: placed on the first device larger than <*minimum volume SYS: size in sectors*> (512 bytes per sector). Additional volumes, named "VOL1", "VOL2", etc., will be created on other devices, one per device.

Example

For one existing device with 50 MB and a NetWare partition, but no volumes:

```
Volumes 51200; 25 MB minimum
```

This command creates volume SYS: on the device and mounts it. For more information on creating and mounting NetWare volumes, see the Volume Management API.

Write_Var_File

Syntax

```
Write_Var_File <path>, <filename>, <varname>, <var value>,
<completion code>
```

Description

Write_Var_File will write the varname and var value to the filename specified by the path/filename. The file is an ASCII formatted file with contents as follows:

"Variable1=true"
"Variable2=false"
"Variable3=Servername"

Example

```
Write_Var_File "sys:system", "varfile.txt", "Variable1",
  "true", ccode
```

NOTE: Ccode will return the file open status. 0 is success, anything else is an error. If the variable to the left contains any spaces, leading or trailing, they will be part of the variable name and must be included with the variable when referencing it later. If the filename doesn't exist, it will be created. The varname and var value will then be written. If the file does exist, the varname and var value will be appended to the contents of the file.

External Command Programming Interface

The external command programming interface contains the functions described below.

IMPORTANT: These functions require ICMD.NLM 2.18 or later.

ICMDSetVar

Use the ICMDSetVar function to set a script variable programmatically.

Function

int ICMDSetVar (char **variableName*, char **variableValue*)

**variableName:* Pointer to name of variable to set
**variableValue*: Pointer to string representing value of variable being set

Return

0 if successful; non-zero error code if unsuccessful.

Description

ICMDSetVar is a function exported by ICMD.NLM that another NLM can call (for instance, in a blocking NLMExec command). This lets the NLM set a variable within a script being executed by ICMD.NLM so the NLM can communicate with the script, changing the control flow, etc.

Example

If MYNLM.NLM has code as follows:

```
ICMDSetVar("mynlmvar", "hello world"); exit();
```

and the script looks like this:

```
NLMExec 1, mynlm
Display 0, "Information: %{mynlmvar}"
```

then this popup text box would be displayed:

```
Information: hello world
```

ICMDGetVar

Use the ICMDGetVar function to get the value of a script variable.

Function

int ICMDGetVar (char **variableName*, char **variableValue*)

**variableName:* Pointer to name of variable
**variableValue*: Buffer containing value of variable; must be at least 128 bytes

Return

0 if successful; non-zero error code if unsuccessful or if variableName does not exist.

Description

ICMDGetVar is a function exported by ICMD.NLM that another NLM can call. This lets the NLM set a variable within a script being executed by ICMD.NLM so the NLM can communicate with the script, changing the control flow, etc.

Example

If a script looks like this:

```
SetVar myvar, "Hello"
NLMExec 1, mynlm
```

and if MYNLM.NLM has code as follows:

```
char buffer[128];
ICMDGetVar("myvar", buffer);
```

then the value in buffer would be the null-terminated string:

```
"Hello".
```

External Command Variable Substitution

A variable, such as the value returned by GetPath, can be used inside any quoted string within an executed external command. The variable is global. It goes out of scope only when the command processor is replaced by a different one. The variable can be dereferenced explicitly by an expression %{<*var name*>} within another string.

Variable	Description
NWSERVER	The name of the server the script is processing on.
NWSRC	Source path where script is found.
NWDST	Destination path (where server will boot from).
NWBOOT	Current path (where server was booted from).
NWLANG	Language number (4 = English, etc.).
COUNTRY	Country ID (DOS country specifier).
CODEPAGE	Codepage number (DOS code page specifier).
OSTYPE	NetWare operating system type: 0 - Native 1 - OS/2 2 - SFT III
LOADERTYPE	Type of loader used to start NetWare: 1 - DOS 2 - OS/2 3 - Windows 3.1
SERVCONFIGTYPE	NetWare server configuration: 0 - Native 1 - SFT III I/O Engine 2 - SFT III MS Engine
SFTLEVEL	Level of active System Fault Tolerance for NetWare.
OSMAJORVERSION	The major version number for the NetWare OS.
OSMINORVERSION	The minor version number for the NetWare OS.

Variable	Description
OSREVISION	The revision level for the NetWare OS.

All path variables, including the predefined ones, allow subfield specifiers in the form <*path var*>.<*specifier*>.

<path var>	*<specifier>*
PATH	String containing the actual path.
VOL	Disk volume name.
PROMPT	Disk volume prompt.
TYPE	Path type.
SUBTYPE	Path subtype.
CONN	Connection number (hex).
ENGINE	Engine number.
TYPE	Values: 1 local server 2 remote server 3 local DOS 4 remote DOS (via RCONSOLE)
SUBTYPE	Values: 0 unknown 1 floppy 2 cd 3 network 4 other

Example

```
NLMExec 1, 'Pinstall %{NWLANG} %{NWSRC.CONN}'
```

If the server language is English, and the connection number for NWRSC is 5, it is as if the following were typed on the system console:

```
'load PINSTALL.NLM 4 5'
```

Installation NLM Command Line Options

Command line options can be preceded by '-', but this is not required.

IMPORTANT: These options are not intended for general distribution; they apply only to the NetWare installation NLM. Other Switches are not guaranteed to be supported in all versions.

User-Documented Switches for NetWare Installation

Switch	Description
DSREMOVE	Allows absolute removal of DS.
R	Allows license replacement.

Switches Helpful for Install Script Writers

Switch	Description
DELAY	Wait until execution of AUTOEXEC.NCF is complete before activating screen.
NOSTATUS	Suppress status bar display in batch mode.
A=	NNNNNNNN:OOOOOOOOOOOO:SSSS source path IPX address: N hex network (8 chars): O hex node (12 chars): S hex socket
B=	Batch file spec.
D=	Destination path in DOS for SERVER.EXE, etc.
E=	Error file spec.
O=	Overwrite existing files flag; 1: always, 2: never overwrite.
S=	Source path DOS specifier.
W=	Overwrite newer files flag; 1: always, 2: never overwrite.

Other Switches

Switch	Description
MDEBUG	Memory debug.
C=	Source path remote context.
F=	Front end flag. Q simplified mode T calculate file set totals V suppressed delayed init (DELAY) X allow more DS options
K=	Default startup.ncf file spec.
L=	Mode of installation.
N=	Source path remote server spec.
P=	Source path remote user password.
U=	Source path remote user name.
Z=	Source path remote tree.

NetWare 4.1 Remote Installation Wrapper

You can use NetWare installation to remotely install a product that was not originally intended for remote installation.

1. Create a wrapper script that copies the associated files to the server before executing PINSTALL.NLM.

   Make sure the system volume is large enough to contain the scratch files, even in local install cases.

   In the example below, the directory layout is as follows on the diskette:

   \ (root)

   PINSTALL.IPS
   PINSTALL.NLM
   other files...

The contents of file PINSTALL.IPS appears as follows:

```
command icmd
getpath dest, 1, 'sys:system\\tmp', "
copyfile 0, 0, 0, 0, 0, NWSRC, '*.*', '', '', dest, '', '', ''
; Additional copyfile commands might be needed to copy all diskettes
; See filedata.doc for details.
nlmexec 1, 'sys:system\\tmp\\pinstall'
erasefile 0, dest, '', '', ''
```

PINSTALL.IPS script causes ICMD.NLM to be loaded. A path variable *dest* is set, all files are copied to that location, PINSTALL.NLM is executed, then the files are erased.

2. Load INSTALL.NLM.
3. Click Product Options > Install a Product Not Listed.
4. Press F3.
5. Enter the remote path:

   <server name>\<vol name>:<path>

NetWare Installation Command Version Availability

The following tables list the available NetWare installation and external installation commands.

NetWare Install Commands

Command	NetWare 4.10	NetWare 4.11	NetWare for Small Business	NetWare 5.*x*
@CDName	X	X	X	X
@DeletePartition	X	X	X	X
@Driver	X	X	X	X
@FileSet	X	X	X	X
@IncludeFile	X	X	X	X
@MessageFile	X	X	X	X
@Other	X	X	X	X
@StartAppObject			X	X
@StartSchemaMod		X	X	X
@SyntaxVersion	X	X	X	X
@TimeZone	X	X	X	X
@TranslateModule	X	X	X	X
@Version	X	X	X	X
Command	X	X	X	X
CopyDriverToServer	X	X	X	X
CopyDriverToStartup	X	X	X	X
CopyToServer	X	X	X	X
CopyToStartup	X	X	X	X
Exec	X	X	X	X

Command	NetWare 4.10	NetWare 4.11	NetWare for Small Business	NetWare 5.*x*
ProductRecord	X	X	X	X
RegisterLanguage	X	X	X	X
SaveLanguageFile	X	X	X	X
Spawn	X	X	X	X

NetWare External Install Commands

Command	Version of ICMD.NLM			
	2.00	2.18	3.20	3.25
Activate		X	X	X
AppendFile	X	X	X	X
CheckFile		X	X	X
Config	X	X	X	X
Console	X	X	X	X
CopyFile	X	X	X	X
CopyFloppyImage	X	X	X	X
Delay	X	X	X	X
DiskReset	X	X	X	X
Display_Text_File				X
Display	X	X	X	X
DSInst	X	X	X	X
EditNCF			X	X
EditNCF_Addline			X	X
EditNCF_Keystrings			X	X

Command	Version of ICMD.NLM			
	2.00	2.18	3.20	3.25
EraseFile	X	X	X	X
ExtractVersionNumbers			X	X
GetDOSServer			X	X
GetPath	X	X	X	X
Goto		X	X	X
GotoIfEqual		X	X	X
GotoIfNEqual		X	X	X
GotoIfGreater		X	X	X
GotoIfLess		X	X	X
GotoIfGrEqual		X	X	X
GotoIfLsEqual		X	X	X
Is_NLM_Active				X
Label		X	X	X
LangVar	X	X	X	X
Menu		X	X	X
NLMExec	X	X	X	X
NLMExec_IO			X	X
Partitions	X	X	X	X
Quit				X
ReadProductRecord			X	X
Read_Var_File				X
SetDir	X	X	X	X

Command	Version of ICMD.NLM			
	2.00	2.18	3.20	3.25
SetDOSServer			X	X
SetLConfigLang	X	X	X	X
SetVar	X	X	X	X
SgotoIfEqual		X	X	X
SgotoIfNEqual		X	X	X
SgotoIfGreater		X	X	X
SgotoIfLess		X	X	X
SgotoIfGrEqual		X	X	X
SgotoIfLsEqual		X	X	X
SpaceCheck	X	X	X	X
SrchNCF			X	X
ValueAdd	X	X	X	X
ValueSet	X	X	X	X
Volumes	X	X	X	X
Write_Var_File				X

A Novell Trademarks

Access Manager is a registered trademark of Novell, Inc. in the United States and other countries.

Advanced NetWare is a trademark of Novell, Inc.

AlarmPro is a registered trademark of Novell, Inc. in the United States and other countries.

AppNotes is a registered service mark of Novell, Inc. in the United States and other countries.

AppNotes is a registered service mark of Novell, Inc. in the United States and other countries.

AppTester is a registered service mark of Novell, Inc. in the United States and other countries.

BrainShare is a registered service mark of Novell, Inc. in the United States and other countries.

C-Worthy is a trademark of Novell, Inc.

C3PO is a trademark of Novell, Inc.

CBASIC is a registered trademark of Novell, Inc. in the United States and other countries.

Certified NetWare Administrator in Japanese and CNA-J are service marks of Novell, Inc.

Certified NetWare Engineer in Japanese and CNE-J are service marks of Novell, Inc.

Certified NetWare Instructor in Japanese and CNI-J are service marks of Novell, Inc.

Certified Novell Administrator and CNA are service marks of Novell, Inc.

Certified Novell Engineer is a trademark and CNE is a registered service mark of Novell, Inc. in the United States and other countries.

Certified Novell Salesperson is a trademark of Novell, Inc.

IPXWAN is a trademark of Novell, Inc.

LAN WorkGroup is a trademark of Novell, Inc.

LAN WorkPlace is a registered trademark of Novell, Inc. in the United States and other countries.

LAN WorkShop is a trademark of Novell, Inc.

LANalyzer is a registered trademark of Novell, Inc. in the United States and other countries.

LANalyzer Agent is a trademark of Novell, Inc.

Link Support Layer and LSL are trademarks of Novell, Inc.

MacIPX is a registered trademark of Novell, Inc. in the United States and other countries.

ManageWise is a registered trademark of Novell, Inc. in the United States and other countries.

Media Support Module and MSM are trademarks of Novell, Inc.

Mirrored Server Link and MSL are trademarks of Novell, Inc.

Mobile IPX is a trademark of Novell, Inc.

Multiple Link Interface and MLI are trademarks of Novell, Inc.

Multiple Link Interface Driver and MLID are trademarks of Novell, Inc.

My World is a registered trademark of Novell, Inc. in the United States and other countries.

N-Design is a registered trademark of Novell, Inc. in the United States and other countries.

Natural Language Interface for Help is a trademark of Novell, Inc.

NDS Manager is a trademark of Novell, Inc.

NE/2 is a trademark of Novell, Inc.

NE/2-32 is a trademark of Novell, Inc.

NE/2T is a trademark of Novell, Inc.

NE1000 is a trademark of Novell, Inc.

NE1500T is a trademark of Novell, Inc.

NE2000 is a trademark of Novell, Inc.

NE2000T is a trademark of Novell, Inc.

NE2100 is a trademark of Novell, Inc.

NE3200 is a trademark of Novell, Inc.

NE32HUB is a trademark of Novell, Inc.

NEST Autoroute is a trademark of Novell, Inc.

NetExplorer is a trademark of Novell, Inc.

NetNotes is a registered trademark of Novell, Inc. in the United States and other countries.

NetSync is a trademark of Novell, Inc.

NetWare is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare 3270 CUT Workstation is a trademark of Novell, Inc.

NetWare 3270 LAN Workstation is a trademark of Novell, Inc.

NetWare 386 is a trademark of Novell, Inc.

NetWare Access Server is a trademark of Novell, Inc.

NetWare Access Services is a trademark of Novell, Inc.

NetWare Application Manager is a trademark of Novell, Inc.

NetWare Application Notes is a trademark of Novell, Inc.

NetWare Asynchronous Communication Services and NACS are trademarks of Novell, Inc.

NetWare Asynchronous Services Interface and NASI are trademarks of Novell, Inc.

NetWare Aware is a trademark of Novell, Inc.

NetWare Basic MHS is a trademark of Novell, Inc.

NetWare BranchLink Router is a trademark of Novell, Inc.

NetWare Care is a trademark of Novell, Inc.

NetWare Communication Services Manager is a trademark of Novell, Inc.

NetWare Connect is a registered trademark of Novell, Inc. in the United States.

NetWare Core Protocol and NCP are trademarks of Novell, Inc.

NetWare Distributed Management Services is a trademark of Novell, Inc.

NetWare Document Management Services is a trademark of Novell, Inc.

NetWare DOS Requester and NDR are trademarks of Novell, Inc.

NetWare Enterprise Router is a trademark of Novell, Inc.

NetWare Express is a registered service mark of Novell, Inc. in the United States and other countries.

NetWare Global Messaging and NGM are trademarks of Novell, Inc.

NetWare Global MHS is a trademark of Novell, Inc.

NetWare HostPrint is a registered trademark of Novell, Inc. in the United States.

NetWare IPX Router is a trademark of Novell, Inc.

NetWare LANalyzer Agent is a trademark of Novell, Inc.

NetWare Link Services Protocol and NLSP are trademarks of Novell, Inc.

NetWare Link/ATM is a trademark of Novell, Inc.

NetWare Link/Frame Relay is a trademark of Novell, Inc.

NetWare Link/PPP is a trademark of Novell, Inc.

NetWare Link/X.25 is a trademark of Novell, Inc.

NetWare Loadable Module and NLM are trademarks of Novell, Inc.

NetWare LU6.2 is trademark of Novell, Inc.

NetWare Management Agent is a trademark of Novell, Inc.

NetWare Management System and NMS are trademarks of Novell, Inc.

NetWare Message Handling Service and NetWare MHS are trademarks of Novell, Inc.

NetWare MHS Mailslots is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare Mirrored Server Link and NMSL are trademarks of Novell, Inc.

NetWare Mobile is a trademark of Novell, Inc.

NetWare Mobile IPX is a trademark of Novell, Inc.

NetWare MultiProtocol Router and NetWare MPR are trademarks of Novell, Inc.

NetWare MultiProtocol Router Plus is a trademark of Novell, Inc.

NetWare Name Service is trademark of Novell, Inc.

NetWare Navigator is a trademark of Novell, Inc.

NetWare Peripheral Architecture is a trademark of Novell, Inc.

NetWare Print Server is a trademark of Novell, Inc.

NetWare Ready is a trademark of Novell, Inc.

NetWare Requester is a trademark of Novell, Inc.

NetWare Runtime is a trademark of Novell, Inc.

NetWare RX-Net is a trademark of Novell, Inc.

NetWare SFT is a trademark of Novell, Inc.

NetWare SFT III is a trademark of Novell, Inc.

NetWare SNA Gateway is a trademark of Novell, Inc.

NetWare SNA Links is a trademark of Novell, Inc.

NetWare SQL is a trademark of Novell, Inc.

NetWare Storage Management Services and NetWare SMS are trademarks of Novell, Inc.

NetWare Telephony Services is a trademark of Novell, Inc.

NetWare Tools is a trademark of Novell, Inc.

NetWare UAM is a trademark of Novell, Inc.

NetWare WAN Links is a trademark of Novell, Inc.

NetWare/IP is a trademark of Novell, Inc.

NetWire is a registered service mark of Novell, Inc. in the United States and other countries.

Network Navigator is a registered trademark of Novell, Inc. in the United States.

Network Navigator - AutoPilot is a registered trademark of Novell, Inc. in the United States and other countries.

Network Navigator - Dispatcher is a registered trademark of Novell, Inc. in the United States and other countries.

Network Support Encyclopedia and NSE are trademarks of Novell, Inc.

Network Support Encyclopedia Professional Volume and NSEPro are trademarks of Novell, Inc.

NetWorld is a registered service mark of Novell, Inc. in the United States and other countries.

Novell is a service mark and a registered trademark of Novell, Inc. in the United States and other countries.

Novell Alliance Partners Program is a collective mark of Novell, Inc.

Novell Application Launcher is a trademark of Novell, Inc.

Novell Authorized CNE is a trademark and service mark of Novell, Inc.

Novell Authorized Education Center and NAEC are service marks of Novell, Inc.

Novell Authorized Partner is a service mark of Novell, Inc.

Novell Authorized Reseller is a service mark of Novell, Inc.

Novell Authorized Service Center and NASC are service marks of Novell, Inc.

Novell BorderManager is a trademark of Novell, Inc.

Novell BorderManager FastCache is a trademark of Novell, Inc.

Novell Client is a trademark of Novell, Inc.

Novell Corporate Symbol is a trademark of Novell, Inc.

Novell Customer Connections is a registered trademark of Novell, Inc. in the United States.

Novell Directory Services and NDS are registered trademarks of Novell, Inc. in the United States and other countries.

Novell Distributed Print Services is a trademark and NDPS is a registered trademark of Novell, Inc. in the United States and other countries.

Novell ElectroText is a trademark of Novell, Inc.

Novell Embedded Systems Technology is a registered trademark and NEST is a trademark of Novell, Inc. in the United States and other countries.

Novell Gold Authorized Reseller is a service mark of Novell, Inc.

Novell Gold Partner is a service mark of Novell, Inc.

Novell Labs is a trademark of Novell, Inc.

Novell N-Design is a registered trademark of Novell, Inc. in the United States and other countries.

Novell NE/2 is a trademark of Novell, Inc.

Novell NE/2-32 is a trademark of Novell, Inc.

Novell NE3200 is a trademark of Novell, Inc.

Novell Network Registry is a service mark of Novell, Inc.

Novell Platinum Partner is a service mark of Novell, Inc.

Novell Press is a trademark of Novell, Inc.

Novell Press Logo (teeth logo) is a registered trademark of Novell, Inc. in the United States and other countries.

Novell Replication Services is a trademark of Novell, Inc.

Novell Research Reports is a trademark of Novell, Inc.

Novell RX-Net/2 is a trademark of Novell, Inc.

Novell Service Partner is a trademark of Novell, Inc.

Novell Storage Services is a trademark of Novell, Inc.

Novell Support Connection is a registered trademark of Novell, Inc. in the United States and other countries.

Novell Technical Services and NTS are service marks of Novell, Inc.

Novell Technology Institute and NTI are registered service marks of Novell, Inc. in the United States and other countries.

Novell Virtual Terminal and NVT are trademarks of Novell, Inc.

Novell Web Server is a trademark of Novell, Inc.

Novell World Wide is a trademark of Novell, Inc.

NSE Online is a service mark of Novell, Inc.

NTR2000 is a trademark of Novell, Inc.

Nutcracker is a registered trademark of Novell, Inc. in the United States and other countries.

OnLAN/LAP is a registered trademark of Novell, Inc. in the United States and other countries.

OnLAN/PC is a registered trademark of Novell, Inc. in the United States and other countries.

Open Data-Link Interface and ODI are trademarks of Novell, Inc.

Open Look is a registered trademark of Novell, Inc. in the United States and other countries.

Open Networking Platform is a registered trademark of Novell, Inc. in the United States and other countries.

Open Socket is a registered trademark of Novell, Inc. in the United States.

Packet Burst is a trademark of Novell, Inc.

PartnerNet is a registered service mark of Novell, Inc. in the United States and other countries.

PC Navigator is a trademark of Novell, Inc.

PCOX is a registered trademark of Novell, Inc. in the United States and other countries.

Perform3 is a trademark of Novell, Inc.

Personal NetWare is a trademark of Novell, Inc.

Pervasive Computing from Novell is a registered trademark of Novell, Inc. in the United States and other countries.

Portable NetWare is a trademark of Novell, Inc.

Presentation Master is a registered trademark of Novell, Inc. in the United States and other countries.

Print Managing Agent is a trademark of Novell, Inc.

Printer Agent is a trademark of Novell, Inc.

QuickFinder is a trademark of Novell, Inc.

Red Box is a trademark of Novell, Inc.

Reference Software is a registered trademark of Novell, Inc. in the United States and other countries.

Remote Console is a trademark of Novell, Inc.

Remote MHS is a trademark of Novell, Inc.

RX-Net is a trademark of Novell, Inc.

RX-Net/2 is a trademark of Novell, Inc.

ScanXpress is a registered trademark of Novell, Inc. in the United States and other countries.

Script Director is a registered trademark of Novell, Inc. in the United States and other countries.

Sequenced Packet Exchange and SPX are trademarks of Novell, Inc.

Service Response System is a trademark of Novell, Inc.

Serving FTP is a trademark of Novell, Inc.

SFT is a trademark of Novell, Inc.

SFT III is a trademark of Novell, Inc.

SoftSolutions is a registered trademark of SoftSolutions Technology Corporation, a wholly owned subsidiary of Novell, Inc.

Software Transformation, Inc. is a registered trademark of Software Transformation, Inc., a wholly owned subsidiary of Novell, Inc.

SPX/IPX is a trademark of Novell, Inc.

StarLink is a registered trademark of Novell, Inc. in the United States and other countries.

Storage Management Services and SMS are trademarks of Novell, Inc.

Technical Support Alliance and TSA are collective marks of Novell, Inc.

The Fastest Way to Find the Right Word is a registered trademark of Novell, Inc. in the United States and other countries.

The Novell Network Symbol is a trademark of Novell, Inc.

Topology Specific Module and TSM are trademarks of Novell, Inc.

Transaction Tracking System and TTS are trademarks of Novell, Inc.

Universal Component System is a registered trademark of Novell, Inc. in the United States and other countries.

Virtual Loadable Module and VLM are trademarks of Novell, Inc.

Writer's Workbench is a registered trademark of Novell, Inc. in the United States and other countries.

Yes, It Runs with NetWare (logo) is a trademark of Novell, Inc.

Yes, NetWare Tested and Approved (logo) is a trademark of Novell, Inc.

ZENworks is a trademark of Novell, Inc.