

VERITAS WinINSTALL™ 7.5

Getting Started Guide

January 2002
30-000623-011


VERITAS

Disclaimer

The information contained in this publication is subject to change without notice. VERITAS Software makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. VERITAS Software shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Copyright

Copyright © 1991-2002 VERITAS Software. All Rights Reserved. VERITAS, VERITAS SOFTWARE, the VERITAS logo, *Business Without Interruption*, VERITAS The Data Availability Company, and VERITAS WinINSTALL are trademarks or registered trademarks of VERITAS Software in the USA and/or other countries. Other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

Printed in the USA and/or Ireland, January 2002.

VERITAS Software
350 Ellis St.
Mountain View, CA 94043
USA
Phone 650-527-8000
Fax 650-527-2581
www.veritas.com



Contents

Chapter 1. Introduction to WinINSTALL	1
User Requirements	2
WinINSTALL Components	2
WinINSTALL Console	2
Discover	2
Distribution Package	3
List File	3
Schedule Files	3
Installers	3
What is New in WinINSTALL 7.5	4
How this Manual is Organized	6
Chapter 2. Installing WinINSTALL: Console and Server	7
Upgrading from WinINSTALL 7.0x to 7.5	8
Installing for the First Time	8
System Requirements	9
Hardware Requirements	9
Software Requirements: Recommended	9
Mapping/Share Requirements	11
Alternative Setups	11
Choosing a Machine for the Console Program	12
Deciding Where Installers Should Reside	12
Running Setup	13
Setting Console Access Levels	16



Installing a Remote Console	17
Chapter 3. Installing WinINSTALL: Preparing Clients	19
System Requirements for the Client Machine	20
Choosing a Delivery Method	20
The Two Basic Installers	20
Other Ways of Running the Automatic Installer	21
Methods for Setting Up the Clients	22
Preparing Users' Machines Through Email	22
Preparing Users' Machines by Modifying the System Logon Script	23
Preparing Users' Machines by Using Centrally Located Files	24
The Automatic Installer	24
Deploying the Automatic Installer	25
The Interactive Installer	26
How the Interactive Installer Works	26
Deploying the Interactive Installer	27
The Scheduler	29
How the Scheduler Works	29
Deploying the Scheduler	30
Customizing the Scheduler	31
The WinINSTALL NT Service	32
How the NT Service Locates Resources	32
Invisibility to the User	33
How NT Service Deals with Accounts	33
Deploying the WinINSTALL NT Service	33
Intranet Installers	35
Deploying the Intranet Installer	36
Creating a Web Server	36
Prepare the HTM Files for Use	37
Making Winstall.htm the Default Document	39



Changing IE Security and ActiveX Settings	39
Configuring WinINSTALL for Netscape Navigator	40
Email Installer	41
Email Installer Best Practices	42
The Reset Partition	42
Deploying a Hidden Reset Partition to Client Machines	43
Customizing the Automatic and Interactive Installers	44
Turning On or Off the Rollback Option	45
Turning On or Off the Uninstall Option	45
Making the Abort Button Visible or Invisible	46
Setting the Post-Installation Pause	46
Setting the Location for Application History	47
Chapter 4. Discover: Creating Distribution Packages	49
The Discover Program	50
Best Practices for Using Discover	51
Run Discover on a Clean PC	52
Create Packages on the Same Platform Your Users Have	52
Clean the Discover Machine Between Runnings of Discover	53
What to Do if You Cannot Use a Clean Reference Machine	53
Taking the Before Snapshot	54
Starting the Discover Wizard for the Before Snapshot	54
Discover Wizard: Welcome Screen	54
Discover Wizard: Selecting Packaging Format	55
Discover Wizard: Selecting Discover Method	56
Discover Wizard: Specify Installer File	56
Discover Wizard: Advanced MSI	56
Discover Wizard: Selecting Target	57
Discover Wizard: List File Selection	58
Discover Wizard: Specifying Work Drive	59



Discover Wizard: Advanced Options	59
Discover Wizard: Drive Selection	60
Discover Wizard: Exclusion File Selection	60
Discover Wizard: Selecting Text Files	61
Discover: Exclusion Registry Selection	62
Choosing Error Suppression	63
Choosing Validation	63
Discover Wizard: Found Files	63
Discover Wizard: Selecting Directory Search	64
Discover Wizard: Specifying Source Directory	64
Discover: Completing the Discover Wizard	65
Recommended Naming Conventions	65
Running the Application's Setup Program	66
Taking the After Snapshot	66
How to Create Distribution Packages Without Discover	67
When to Customize Distribution Packages	68
Problems That Can Occur with Shared DLLs	69
NwCalls.dll	69
Winsock.dll	69
Avoiding Trouble When Uninstalling Applications	69
Testing a Distribution Package	70
Command Line Arguments for Discover	70
Exclusion Files	71
Tech Note: OS Image and Reset Partition from Boot Floppy	72
Chapter 5. Editing Distribution Packages: The Basics	73
When to Customize Distribution Packages	74
How to Edit a Distribution Package	75
Index	77



VERITAS WinINSTALL automatically distributes software, configuration changes, updates, upgrades, and replacement data files rapidly and completely to any group of workstations within a local, wide area, or enterprise network.

The system administrator, for example, installs a single copy of a software program on a single workstation. WinINSTALL's Discover program creates a package containing all changes made to the workstation's configuration in order to accomplish the successful installation. The package also contains any additional files required for the installation. This package is stored for deployment. The actual processing of the package is either "push", controlled by the administrator, or "pull", controlled by the user.

The system administrator can customize packages and track the status of deployed packages using the console program. No scripting is necessary.

The methods used to create and distribute these packages provide the system administrator with enormous flexibility and the ability to impose consistency. In addition to deployment of software and upgrades, benefits include:

- ◆ Reversal of installation or uninstallation
- ◆ Distribution of patches, data files, printer drivers, etc.
- ◆ Modification of scripts and configuration files globally, restricted to a group, or even individually
- ◆ Modification of the desktop
- ◆ Removal of unauthorized software

The ease of use and rapid deployment means that WinINSTALL reduces the time normally required for software maintenance in a large network and lowers the total cost of ownership for most applications.



User Requirements

The WinINSTALL administrator is usually the system administrator. You must have security access to the network. For example, you can be a member of the local administrators group. In addition, you should understand the basics of Windows NT security. You must have a solid understanding of your network, the network operating system, and the appropriate workstation operating systems.

Some optional operations require an understanding of the Windows registry.

For certain complex installations some users prefer to create a script. This is usually unnecessary, but a scripting language, Visual Basic Script, has been included. If you intend to use this option, you must know BASIC. You will find reference materials for the scripting language on the CD-ROM in Acrobat format:

- ◆ Lr225.pdf (Language Reference)
- ◆ Ug225.pdf (User's Guide)

WinINSTALL Components

WinINSTALL is not a single program, but several components working together to create and deploy installation packages.

WinINSTALL Console

The WinINSTALL console is the user interface for WinINSTALL. Use it to edit distribution packages, to create operating system templates, and to manage certain features of the WinINSTALL program.

Discover

The Discover program (Disco32.Exe) is a standalone program that acts like a camera. It takes a "snapshot" of a workstation before an installation is started. Once the installation is complete, Discover takes a second snapshot.

The program compares the two snapshots and finds the differences between the two. This includes not only the new files, but changes to configuration files, the Registry, etc.

All of these changes and the instructions for accomplishing them are copied into a distribution package.

Distribution Package

A distribution package is a file containing the instructions, changes, and files that make up a successful installation of the product you want to distribute. Distribution packages are stored in an area accessible to all users. There are two types:

- ◆ NAI package, sometimes called WinINSTALL installer packages, are distribution packages created only through WinINSTALL's Discover program or through the WinINSTALL Console. They have the .nai file extension.
- ◆ MSI packages, sometimes called Windows Installer packages, can be created by WinINSTALL's Discover, but they have a different structure than WinINSTALL packages. They have the .msi file extension.

Distribution packages can be installed individually, or as part of a group of products defined in a list file.

List File

List files contain sets of distribution packages. They are commonly used to deploy products that will be installed on a definable group of workstations. For example, you may want to send a different set of programs to certain workstations based on the type of work, the type of display, or the remaining free space on the hard disk drive. List files use the extension .lst.

You can set up any number of list files, but you should base them on the way you want to manage the system. If you need the option, you can create list files that contain other list files.

Schedule Files

The administrator creates schedule files (.sch) from the console. These files are optional. They are used when the administrator uses the automatic installer and needs to be specific about a time frame during which the program will be installed on specific workstations.

Installers

Installers are the programs which process the distribution package or list. They are stored in an area accessible to all users. The installers provide the following installation options:

- ◆ Automatic – push distribution that lets you install the program automatically without user intervention. Push distribution can be “quiet”: installing the new materials without informing the user.



- ◆ WinINSTALL Scheduler – push distribution that lets the WinINSTALL administrator select a date and time to activate the installation.
- ◆ NT Service – secure push distribution using encrypted passwords.
- ◆ Interactive – pull distribution that lets the user control the timing of the installation.
- ◆ Email – pull distribution that distributes a self-installing package attached to an email message for user activation.
- ◆ Intranet – pull distribution by placing the self-installing package on an intranet web page for user activation with ActiveX or Netscape plug-in.

Windows 2000 has its own installer, called the Microsoft Windows Installer, which can install MSI packages but not NAI packages. There are also versions that can be installed on 9X and NT 4 clients if users want to deploy MSI packages to those clients. For these clients, version 1.1 must be installed.

What is New in WinINSTALL 7.5

The latest version of WinINSTALL has a number of new and enhanced features:

- ◆ Enhanced Microsoft® Windows Installer (MSI) Support and Transparency

WinINSTALL 7.5 strengthens what we offer in-terms of MSI package creation and editing. In addition, recognizing that MSI packages are complex, we take care of the complexity in the background and make package creation and editing for MSI seem like creation and editing of the simpler WinINSTALL Network Application Installer (NAI) files. The customer does not need to be an MSI expert; he or she just needs to know how to use WinINSTALL.

- ◆ Support for Transforms (.mst files) and Patches (.msp files)

You can now create transforms through the Discover program, and you can create patches through the Patch Wizard.

- ◆ Hidden Reset Partition for Automated System Reset

This is a workstation availability (disaster recovery) feature that eliminates the need for boot media (i.e. boot floppy or boot CD.) It allows automatic, unattended restore of the workstation operating system before recovering the applications and data.

- ◆ Replication Error Checking

WinINSTALL provides basic replication functions, but packages often have parameters that are local to one server and cause problems for replication. We cannot fix these problems because they are environmental. Therefore, we provide a method for checking the package before replication and alerting the administrator about problem areas so he or she can correct them.

- ◆ **Third-party Support**

WinINSTALL has basic operating system installation/restore capabilities, but we realize that many customers have third party processes (such as Ghost) that are used and will continue to be used for staging. Third-party support is a command line option that allows users to launch the operating system staging tool or process of their choice for this part of the process.

- ◆ **Load Set CD Creation Wizard**

A key strength of WinINSTALL is the ability to distribute software load sets to non-networked and remote machines. We provide a CD Creation Wizard to lead the users through the necessary steps to make a CD for the installation package source.

- ◆ **Internet Distribution, Including Check Point Restart**

Check Point Restart is the process of pulling down all the necessary files to the target system's local disk before initiating the install. If the connection is broken before all of these files can be staged locally, the staging process restarts where it left off. This function is designated for all URL-referenced files (i.e., <http://something> file source (as opposed to UNC path or network share.)

- ◆ **Digital Signature Support**

Digital Signature Support is a security feature. It is the ability to embed into the package a certificate of authenticity, along with checksum or Cyclic Redundancy Check (CRC) data on the package itself. The recipient can be assured that no one has tampered with the package.



How this Manual is Organized

As much as possible, this manual is task oriented. After a brief introduction -- what WinINSTALL is, how it is used, what this version offers that is new -- we discuss the four following tasks:

- ◆ Installing WinINSTALL, server side and client side
- ◆ Creating distribution packages, a task performed outside the console
- ◆ Creating operating system templates
- ◆ Editing distribution packages or operating system templates

This fourth task, however, encompasses hundreds of possible concepts and functions. To help the new user find what he or she wants to know, the manual is organized based on the eight list-view areas:

- ◆ General
- ◆ Files
- ◆ Shortcuts
- ◆ Registry
- ◆ Services
- ◆ Edits
- ◆ Advertising
- ◆ Patches and Transforms

Installing WinINSTALL: Console and Server

2

This chapter discusses upgrades and new installs. Upgrading merely requires that you uninstall the old version and install the new one. The technique is discussed very briefly at the beginning of the chapter.

New installation is divided into two portions:

- ◆ Installation of the WinINSTALL console and the network files
- ◆ Installation of the client portions (See “Installing WinINSTALL: Preparing Clients.”)



Upgrading from WinINSTALL 7.0x to 7.5

Upgrading from WinINSTALL 7.00x to 7.5 requires only two steps:

- ◆ Confirm that you meet the system requirements for the new version.
- ◆ Run Setup to uninstall the older version, including the replication service (RepSvc). To uninstall Replication, login to the network share machine and uninstall Replication using the Add/Remove Programs option in Control Panel.
- ◆ Run Setup again to install the new version of WinINSTALL.

▼ **To uninstall the old version:**

1. Run Setup.
2. When the Welcome screen is displayed, click **Next**.
3. When the Program Maintenance screen is displayed, click **Remove**.
4. When the Remove the Program screen is displayed, click **Remove**.

The old WinINSTALL packages are not touched during the uninstall, and they convert automatically. The old database is compatible with the updated version of WinINSTALL, and the replication database is left intact.

Installing for the First Time

The process of installing WinINSTALL for the first time consists of the following steps:

Steps to Installing WinINSTALL

Step	Process
1	Examine system requirements, including the section on Alternative Setups.
2	Select a console machine.
3	Decide where you want the WinINSTALL installer programs to reside. You will need to provide this information during Setup. By default, the area where you install the installers will also be where you will put distribution packages that you create later on.
4	Install console and server by running Setup from the selected console machine.

Steps to Installing WinINSTALL

Step	Process
5	Install clients components (See "Installing WinINSTALL: Preparing Clients".)

System Requirements

Plan ahead when choosing the server on which you want to store distribution packages. By default, this will be the same machine on which you place the installers. This machine will need a good deal of space to accommodate distribution packages, which can be very large.

Hardware Requirements

- ◆ Server

The server is defined as the machine that holds the \WINSTALL directory. It may be the same machine you use for the Operating System distribution site. Allot 168 MB of hard disk space for WinINSTALL program files. Plan ahead to have enough space to accommodate the distribution packages you will create. In addition, if the server will also be used as an Operating System distribution site, you should allot storage space for those files as well.

- ◆ WinINSTALL Console

- Disk space: 225 MB
- Display resolution: 800 by 600 or better

Software Requirements: Recommended

This area discusses the software requirements for complete functionality of WinINSTALL version 7.5. The section called Alternative Setups discusses the possibilities and limitations of setups that do not meet these recommendations.

- ◆ WinINSTALL Console: Recommended

- Windows 2000 or XP, which by default includes the Windows Installer (If you wish to install the console on another platform, see Alternative Setups for special requirements.)



- Users who wish to build packages specifically for versions 1.2 and 2.0 of MSI must install the desired version of MSI on the console machine.
- IE 5.0 or greater

Note Setup doesn't run if Norton Anti-Virus is running.

◆ **Discover Client Machine: Recommended**

The Discover client machine is the machine on which you run Disco32.EXE to create distribution packages. The two most important requirements for this machine are:

- It must be a clean machine.
- It should use the same platform as the users who will process the package you are creating.

If you want to use Discover to create NAI packages, no special requirements apply.

If you want to use Discover to create MSI packages, the Windows Installer must reside on the Discover client machine. For Windows 2000 or XP machines, this installer is part of the operating system. If the machine is NT4, you will need to install the Windows Installer yourself. The program is available in the MSI directory on the CD. Click on it to run and install it.

Users who wish to build packages specifically for versions 1.2 and 2.0 of MSI must install the desired version of MSI on the Discover machine.

Note We cannot Discover applications that have been installed by the Windows Installer.

Note If the Discover client machine is Windows 9x, you can create NAI packages. However, you cannot create an MSI package directly with Discover. On Windows 9x machines, you should create an NAI package and convert it to MSI format by opening the **Actions** menu and selecting **Generate MSI**.

◆ **WinINSTALL Server: Recommended**

We recommend that the WinINSTALL Server be a Windows 2000 or XP machine. If you wish to use a machine with another operating system, see Alternative Setups for special requirements.

◆ **Installers: Recommended**

The Installers will run on any of the following systems:

- Windows 2000 or XP
- Windows 95/98
- Windows NT 4.0

For full functionality (i.e., the ability to install both MSI and NAI packages), clients must have the Windows Installer version 1.1 or higher on their machines. Windows 2000 or XP provides this as part of the operating system. Clients with NT 4 and 9x must install the Windows Installer themselves.

Mapping/Share Requirements

In order to install the network components of WinINSTALL:

- ◆ Make sure you have a valid share path (local or networked).
 - Make sure you have an authenticated connection to the share/volume by directly accessing the share prior to installation.
 - During installation, even if you have a mapped drive, you should install to the UNC for the share.
- ◆ Be logged on as system administrator to run Setup.

Alternative Setups

This section describes software requirements for those who do not need full functionality of WinINSTALL.

WinINSTALL Console: Alternative

If you set up the console on NT 4.0, we recommend Service Pack 6 or greater. If you do not know what Service Pack you have, open the Start menu and select **Programs**. Select **Administrative Tools** and then **Windows NT Diagnostics**. Service Pack information is on the **Version** tab.

Note If you run Setup on a machine with less than Service Pack 5, you will see a message indicating that the share machine does not have the sufficient Service Pack level. Installation will continue, but you should go back and add Service Pack 5 later.

If you wish to manipulate MSI packages on NT4 systems, you must install the Microsoft Windows Installer on the console machine. The program is available in the MSI directory on the CD. Click on it to run and install it. Any menu options relating to MSI packages will be grayed out on NT 4 systems until the Windows Installer is installed.

Internet Explorer version 5 should be installed on the console machine. This is available through Microsoft's web site.



If you are installing onto an NT system, it is best to use a domain account for the replication service logon account. When running the setup program for the remote console, you should be in the same domain as the WinINSTALL server to avoid any rights conflicts.

WinINSTALL Server: Alternative

The WinINSTALL server can be one of the following:

- ◆ A Windows NT4 machine with Service Pack 5 or Service Pack 6a or greater
- ◆ A Novell volume (but replication will not be functional)

The Microsoft Windows Installer should be installed on the machine that holds the \WINSTALL directory. The program is available in the MSI directory on the CD. Run it to install the Windows Installer.

Replication will not install to a Novell server.

Mapping Requirements for Alternative Setups

If using an Novell server as your server machine, map a drive to where your WINSTALL directory will be; just authenticating is not sufficient.

Choosing a Machine for the Console Program

Before running Setup, you must choose a console machine. Keep the following points in mind:

- ◆ The WinINSTALL Console program can only be run on a Windows 2000/XP machine or on NT 4.0 (with Internet Explorer 5.0 and NT Service Pack 5 or 6a installed).
- ◆ If you wish to manipulate MSI packages, the Microsoft Windows Installer should be installed on the console machine. Windows 2000/XP provides this as part of the operating system. For NT 4.0, you must install it yourself.
- ◆ The console machine should be in the same domain as the server that will hold the WINSTALL directory and the replication service files.

Deciding Where Installers Should Reside

Before running Setup, you should decide where you will want the installers to reside. By default the name of the directory is WINSTALL. In choosing the location of the installers, keep the following in mind:



-
- ◆ The selected area must be accessible to you when you run Setup.
 - ◆ It must be accessible to your users when they process distribution packages. This is because users do not actually have installer programs on their own computers. If they cannot access the installers, they will be unable to process distribution packages.
 - ◆ If you are in an NT environment, you may want to place the installers onto a shared drive. More commonly, however, people select the first networked drive. This is the default if you select a Typical setup.

Note By default - and with our *strong* recommendation - distribution packages will be placed on the machine that holds the installers. Therefore, this machine should have a good deal of storage space, since distribution packages can get very large.

Running Setup

After choosing a console machine and deciding where you want the WinINSTALL installers to reside, run Setup.

Note In the unlikely event that installation does not start because the installation engine is corrupt and missing files, delete the @CommonFiles\InstallShield directory and re-run Setup.

▼ To run Setup:

1. Before running Setup, stop all services associated with Norton Anti-Virus.
2. Insert the VERITAS Software CD into the CD-ROM drive of the machine you have chosen.
3. The Choose Setup Language dialog appears. Choose a language from the drop-down list: English, French, or German.
4. Click **OK**.
5. When the Welcome screen appears, click **Next** to begin setup.
6. Complete the License Agreement screen:
 - a. Read the license agreement:
 - If you accept, select **I accept...** and click **Next** to continue setup.
 - If you do not accept, select **I do not accept...** and click **Cancel** to exit the wizard.



- b. Click **Next**.
7. Complete the Customer Information screen.
- a. Fill in your name and company, if they do not already appear. Then do one of the following:
 - If installing an evaluation copy, select **Evaluation**. The software evaluation period is for 30 days and cannot be extended.

Note If you choose *not* to add a license number when running Setup, and instead add the license number after installation via the console, then `Brander.exe` must be run separately in order to license the Installer and Discover programs. (`Brander.exe` resides in the `WINSTALL` directory.) If not, the installers and Discover will cease to work after 30 days since they will still be branded with only the original 30 Day Demo information.

- If installing a licensed copy, click **Next** or **Add License** to bring up the **Add License Key** dialog. Type the license key and click **OK**. Then click **OK** again.
- b. Click **Next**.
8. Complete the Destination Folder screen:
- a. Using the following guidelines, select the local installation directory.
 - The local installation directory is where you will install the local components of WinINSTALL.
 - If you do not want to install the console components in the default directory, click **Change** and locate the folder you want.
 - Do not use a UNC path for the local drive.
 - b. Using the following guidelines, select the network installation directory.
 - This is where the installers will reside. By default, it is also where distribution packages will be created.
 - Use the UNC path for the network installation directory.
 - Do not use the `C:\Program Files` area unless this is a *shared* drive accessible to all users who will need to run the installers. (Read “Deciding Where Installers Should Reside” on page 12 if you have not done so already.)
 - If you get a warning message that the drive selected is not a networked drive, you can share it out after the install as long as the share name matches the volume or folder name.

-
- If you want to distribute or replicate MSI packages, the Windows Installer must be installed on the volume specified for the Network Drive. This installer can be installed after Setup, if you choose.
- c. Click **Next**.
9. Complete the Setup Type screen:
- a. Indicate whether you want to perform a typical or a custom setup.
If you select custom setup, you will need to complete the Software Distribution Components screen:
 - Select individually the components you want to install.
 - Click **Next**.
 - b. Click **Next**.
10. Complete the Operating System Distribution Setup screen:
- a. Read the recommendations on the screen regarding the base distribution site.
 - b. Select the location for the site.
 - c. Click **Next**.
11. Complete the WinINSTALL Replication Service Setup screen:
- The WinINSTALL replication service allows WinINSTALL packages to be replicated from one WinINSTALL server to another.
- a. Type or browse for the user name of the account you want to use to run the replication service.
-
- Note** The account must be a member of the administrators group on the machine with the network folder. In addition, if you want to distribute or replicate MSI packages, the Windows Installer must be installed on the volume specified for network drive.
-
- b. Type and confirm the password of the account you want to use to run the replication service.
 - c. Click **Next**.
12. If you are satisfied with the settings you have made, click **Install**. To make changes, click **Back**.



13. After the local machine has finished rebooting, reboot the network machine to be able to run the replication service.

Note Your next task is to install the client components of WinINSTALL. See “Installing WinINSTALL: Preparing Clients”.

Caution After installing WinINSTALL, do not rename the machines specified during the WinINSTALL Setup process.

Setting Console Access Levels

As administrator, you may be the only person you want to have access to the WinINSTALL console. Once you have finished Setup, you can grant different users or groups on your network different access levels.

Note You can work with WinINSTALL without setting this type of security. If you do decide to set Console access levels, however, the task should be done by an administrator. Be sure that you are part of the group you add. Otherwise you will lock yourself out of using the Console for WinINSTALL.

▼ **To set console access levels:**

1. Open the Console and highlight **Software Distribution Lists**.
2. Open the **Configure** menu and select **Security**.
The Access dialog will be displayed.
3. Expand any of the branches and sub-branches.

For the highlighted element, a list of users and groups will be displayed in the right-hand window.

Note With Novell client installed, if you select NDS when trying to add a user or group to the security section, under certain conditions no users or groups appear. To make sure that the NDS area is populated, have the latest Novell client32 installed. Be a user with Admin rights to a Novell server with the WinINSTALL share, and be logged into and mapped to the Novell WinINSTALL share.

4. Right-click on the group or user for whom you want to set access, and select **Properties**.

5. Select an access level.

You can assign different rights to different users/groups. Once you have assigned these levels for a particular user or group, that access level will apply across management platforms for that user or group.

The following access levels are available:

- None - The specified users cannot access the list files.
- Read-only - The specified users can see but cannot edit the list files.
- Modify - The specified users can edit but not add or delete distribution packages.
- Create - The specified users can edit or add to list files, but not delete.
- Full - The specified users can edit, add, or delete packages and list files.

6. Click **OK** when you have assigned all access levels you want to assign.

Installing a Remote Console

If you want more than one system administrator to be able to edit distribution packages, you can set up a second (or third) console machine. We refer to this as the remote console.

If you are installing the remote console onto an NT system, it is best to use a domain account for the replication service logon account. You should be in the same domain as the WinINSTALL server to avoid any rights conflicts.

▼ **To set up a remote console:**

1. Go to the machine where you want to install the remote console.
2. Navigate to the shared `wininstall` directory where the installers and NAI packages reside.
3. Run `WnAdm32.Exe`.
4. Reboot even though you will not see a message requesting you to do so.
5. If you receive an error message indicating that replication has not been installed, click **OK** to make the error message disappear.
A status bar will indicate that installation is occurring.
6. Reboot the computer.



You will be able to modify distribution packages from this console, although you should try to avoid having separate administrators editing the same packages at the same time so that there is no conflict in what version is being saved.

Installing WinINSTALL: Preparing Clients

3

Before preparing WinINSTALL clients, you must install the console and server components of the product. If you have not done so, see “Installing for the First Time.”

Preparing users' workstations consists of making certain changes to their registry so that they can use one or more WinINSTALL installers. For several installers, WinINSTALL provides distribution packages to automate this process.

Preparing the user's machines consists of the following steps:

- ◆ Confirm system requirements
- ◆ Choose a delivery method
- ◆ Read about the installer you have chosen
- ◆ Deploy the installer
- ◆ (Optional) Customize the behavior of the automatic and interactive installers
- ◆ (Optional) Create a hidden reset partition on the client workstation to be used for re-initialization if necessary in the future



System Requirements for the Client Machine

The Installers will run on clients with any of the following systems:

- ◆ Windows 2000/XP
- ◆ Windows 95/98
- ◆ Windows NT 4.0

Clients must have access to the share that holds the WinINSTALL installer you want them to use.

For full functionality (i.e., the ability to install both MSI and NAI packages), clients must have the Windows Installer version 1.1 on their machines. Windows 2000/XP provides this as part of the operating system. Clients with NT 4 and 9x must install the Windows Installer themselves.

Choosing a Delivery Method

There are two types of installers: automatic and interactive. The automatic installer can also run in four enhanced modes:

- ◆ As a scheduler
- ◆ As an Email delivery system
- ◆ As an intranet installer
- ◆ As an NT service

The Two Basic Installers

Your users must be set up with access to at least one of these installers; they can use both.

Installer	How It Works	Advantages and Disadvantages
Automatic Installer (Winstala.exe)	Can be configured to run automatically at Windows startup. Packages are automatically processed at that time.	Gives the administrator primary control over the PC environment because users do not choose which distribution packages to process.

Interactive or Menu-Driven Installer (winstall.exe)	Lets users see and choose from a menu of available installations.	Gives users a good deal of control over their PC environment since they can choose when and whether to process a package.
---	---	---

Other Ways of Running the Automatic Installer

Users must have access to the automatic installer in order to use any of these enhancements.

Mode	How It Works	Advantages and Disadvantages
As Scheduled Installer (Sswsched.exe)	Processes distribution packages during a scheduled time frame by activating the automatic installer.	Allows the administrator great flexibility in deciding who shall process a distribution package and when. Although users can reschedule a job (if you permit them to do so), they cannot refuse to process a package, as they can with the interactive installer.
Through Email	Sends the desired distribution packages as attachments to designated users. When they receive the mail, the users activate the automatic installer by double-clicking (or whichever method your own email package uses) the attached icon.	Users have control over when and whether to initiate the installation, uninstallation, or upgrade.
As Intranet Installers	Allows you to place distribution packages on an intranet web site.	Administrator can include other information on the web page. User must have access to WinINSTALL.



Mode	How It Works	Advantages and Disadvantages
As NT Service	Provides unattended operation of the automatic installer on Windows NT clients and servers.	Runs in background and doesn't present an interface to the user. Processes distribution packages whether or not the user is logged on to the PC. The NT service must be installed on the client workstation for this method of distribution to operate.

Methods for Setting Up the Clients

Whether or not you want your clients to use the automatic installer in the future, they must run the WinINSTALL automatic installer one time to make the necessary changes on their machines. You can use any of the three following techniques to force this run:

- ◆ **Email method**
Send each user a batch file through email, and tell the users to run the file.
- ◆ **Login-Script Method**
Modify your login script to execute the batch file.
- ◆ **Direct Access Method**
Put `WiRun.exe` or `Winstala.exe` in a directory accessible to everyone, and instruct your users to run the appropriate file.

Use whichever method is appropriate to your particular network. You generally do not need to use more than one.

Preparing Users' Machines Through Email

If your network does not use logon scripts - or if you do not want to modify the logon script - you can prepare users' workstations by sending them email which includes, as an attachment, a shortcut to either of the following files, which they will run once:

- ◆ `WiRun.exe`, which prepares workstations that log into the network before starting Windows
- ◆ `Winstala.exe`, which prepares workstations that log into the network after starting Windows

The `WIRun.exe` and `Winstala.exe` files must be located in the `Wininstall` directory accessible to all users.

Preparing Users' Machines by Modifying the System Logon Script

If you are in an environment that uses either NetWare or Windows logon scripts, you can prepare users' workstations by modifying the logon script so that users run either of the following files once:

- ◆ `WIRUN.EXE`, which prepares workstations that log into the network before starting Windows
- ◆ `WINSTALA.EXE`, which prepares workstations that log into the network after starting Windows

Place one of these programs into a batch file that will be run by the logon script.

Creating the Batch File

▼ Use the following guidelines for creating the batch file:

1. Create a batch file that includes a UNC path to the `WINSTALL` directory and an instruction to run either the `WINSTALA.EXE` program or the `WIRUN.EXE` program. For example:

```
\\server name\ShareOrVol\WininstallDirectory\WINSTALA.EXE
```

2. Replace "server name" with the actual name of the server on which WinINSTALL was installed. Replace `ShareOrVol` with the actual share name or volume name. Replace `WininstallDirectory` with the actual directory where you placed the automatic installer.

How to Modify the NetWare Logon Script

▼ To modify the NetWare Logon script:

1. Decide which logon script to modify:
 - Container login scripts run first and set environments for all users in a container.
 - Profile login scripts run after container login scripts. They set environments for multiple -- but not necessarily all -- users.
 - User login scripts run last. They set the environment for a single user.



It may be easier to create a Profile Logon Script because this type of script applies to users in all NDS containers and executes before the users' individual logon scripts.

2. Modify the last line of the logon script to run the batch file. The last line must be the `EXIT` command. Add the name of the `.BAT` file in quotation marks after the `EXIT` command. The name must be 14 characters or less. For example:

```
Exit "WiRun.Bat"
```

How to Modify the Windows NT or Windows 2000 Logon Script

The logon script must be located in the following path:

```
\Winnt\system32\Repl\Import\Scripts
```

The file located here can be created in Notepad (or any word processor), but must have a `.CMD` extension. Add the `.CMD` file to the user logon through the User Manager by adding the Logon Script Name to either the user or the group.

Preparing Users' Machines by Using Centrally Located Files

If your network does not have Email capability or use logon scripts, or if you do not want to modify the logon script, you can prepare users' workstations by asking users to run either of the following files once:

- ◆ `WIRUN.EXE`, which prepares workstations that log into the network before starting Windows
- ◆ `WINSTALA.EXE`, which prepares workstations that log into the network after starting Windows

The `WIRUN.EXE` and `WINSTALA.EXE` must be located in the `WINSTALL` directory accessible to all users.

The Automatic Installer

The automatic installer (`winstala.exe`) runs distribution packages automatically at Windows startup for clients who have been set up to run that installer. The program runs from the network or a shared drive; users do not have their own individual copy of an installer on their local drives. What they do have is a line in their `win.ini` file or their Registry telling them to run the automatic installer or its stub file.

When it runs, the automatic installer follows these steps:

Step	Process
1	The automatic installer, by default, consults the <code>WinApps.lst</code> file to see which distribution packages have been made available for install or uninstall. It also checks every package for access restrictions.
2	It checks the user's <code>Nai.ini</code> or registry to see if any of the available packages have already been installed.
3	It cycles through the list, processing each of the appropriate distribution packages in turn. It follows each file's instructions, which may be minimal or complex, depending on the application and the ways the administrator may have modified the distribution packages. The installer will skip any package that has already been installed, that cannot be read, or that is not accessible to the user because of access restrictions or unmet prerequisites.
4	If an application is installed, WinINSTALL notes that fact in the registry or the <code>Nai.ini</code> file. This information becomes part of the application history. If you have configured the database and turned on log file notification, the information can also be stored in database tables.

Deploying the Automatic Installer

The WinINSTALL automatic distribution package, in the WINAPPS list, automates deployment of the automatic installer.

▼ To deploy the automatic installer:

1. Open the console, and highlight the WinINSTALL Automatic package.
2. Expand the package, which is compressed:
 - a. Open the **Actions** menu.
 - b. Select **Expand**.
 - c. Click **Yes** to confirm.
3. After the package is expanded, turn on **Allow Access**:



- a. Click the Conditions tab and then the General tab.
 - b. Select **Allow Access**.
 - c. Resave the package.
4. Ensure that clients run the automatic installer once to process the package. Use any of the following methods:
 - Email method
 - Logon modification method
 - Central-files method
 5. (Optional) You may also add switches and parameters to the command line. (See “More on Installers.”)

The Interactive Installer

The interactive installer (`winstall.exe`) gives the user a choice of when and whether to run distribution packages. When the user clicks the **Interactive Installer** icon, a list of applications available for installation or uninstallation appears. The user can select one or more of them or can ignore the list entirely.

Note By default, the interactive installer uses the same list file as the automatic installer (`winapps.lst`). If the user has both the automatic installer and the interactive installer, the automatic installer will process the distribution package. If you activate both installers, we recommend that you create a separate list file for the interactive installer.

How the Interactive Installer Works

The interactive installer works as follows:

Steps for Using the Interactive Installer

Step	Process
1	The user clicks on the Interactive Installer icon, and the installer consults the designated list for application packages available to be processed.

Steps for Using the Interactive Installer

Step	Process
2	The user chooses one or more packages to process. Based on WinINSTALL security features set by the administrator, some packages on the list may not be available to the user.
3	WinINSTALL displays a check beside any package that has already been installed. Once the user clicks Install , a confirmation dialog listing all selected packages appears, allowing the user to continue or cancel. The date and time of the last installation appear next to any previously installed packages.
4	The installer follows each file's instructions, which may be minimal or complex, depending on the application and the ways the administrator may have modified the distribution package. During installation, a completion bar indicates the progress of the procedure.
5	If an application is installed, WinINSTALL notes that fact in the registry or the <code>Nai.ini</code> file. This information becomes part of the application history. If you have configured the database and turned on log file notification, the information can also be stored in database tables.

Deploying the Interactive Installer

The WinINSTALL Interactive User Menu package, in the Winapps list, automates deployment of the interactive installer.

▼ To deploy the installer:

1. (Recommended) Create a special list for interactive installer packages. This is particularly important if some of your clients will be using the automatic installer.
 - a. Start the Console and highlight **Software Distribution Lists**. (Do not highlight Winapps because doing so would make your new list a child of Winapps.)
 - b. From the **File** menu, select **New**, or right-click and select **Add List**.
 - c. Type in or browse for the list you want to add.
 - d. In the **Description** field, type the text you want to appear in the tree view list.
 - e. Click **OK** to create an empty list file to which you can add distribution packages.



2. Expand the Interactive User Menu package, which is compressed:
 - a. Open the **Actions** menu.
 - b. Select **Expand**.
 - c. Click **Yes** to confirm.
3. If you are using the recommended separate list, instruct the interactive installer to use that list:
 - a. With the package highlighted, select **Shortcuts** in the list view window.
 - b. Double-click on the shortcut that is listed.
 - c. In the Target File field, give the UNC path and file name of the interactive installer, followed by the name of the list file. Use the following syntax:

```
@WINSTALLDIR\winstall.exe interactivelist.lst
```
 - d. Click **OK**.
4. Modify the package to turn on **Allow Access**.
 - a. Select **General** in the list view window.
 - b. Click the **Summary** tab and then the **General** tab.
 - c. Turn on **Allow Access**.
 - d. Save the package.
5. Ensure that clients run the automatic installer once to process the package. Use any of the following methods:
 - Email method
 - Logon modification method
 - Central-files method

The Scheduler

The WinINSTALL scheduler lets you set a window of time during which to process a distribution package. The scheduler does not perform installations itself. Instead it starts the automatic installer, and the automatic installer does the work.

When you schedule packages, you can create very specific recipient lists, indicating who should process the package. During the designated time period, the automatic installer processes the package for the designated users or groups. You have the option of allowing the user to reschedule the job to another time.

How the Scheduler Works

As administrator, you can tell the scheduler how often to run. (See “Customizing the Scheduler” on page 31.) By default the scheduler checks for jobs every 60 minutes. It goes through the following steps:

Steps for Using the Scheduler

Step	Process
1	At designated intervals, the scheduler looks for schedule files (.sch) in the main WinINSTALL directory.
2	It examines the user's <code>Nai.ini</code> or registry and compares the date and time of the Last Schedule Run with the date/time stamp on any .sch file it finds.
3	If a .sch file is newer than the Last Schedule Run, the scheduler starts the automatic installer.
4	The automatic installer checks whether or not the user is a targeted recipient for the job. If not, it exits without the user ever knowing anything happened.
5	If the user is a targeted recipient, he or she may be given a chance to reschedule the job. (As administrator, you decide whether this option is to be available for a particular distribution package.) Otherwise, processing of the distribution package occurs immediately.
6	The automatic installer processes the distribution package.



Steps for Using the Scheduler

Step	Process
7	The automatic installer places the current date and time into the Last Scheduled Run field of the <code>Nai.ini</code> file or the registry. It also updates the Application History information in the <code>Nai.ini</code> file or registry.

Deploying the Scheduler

The WinINSTALL Scheduler package, in the Winapps list, automates deployment of the scheduler.

▼ **To deploy the installer:**

1. Open the console, and highlight the WinINSTALL Scheduler package.
2. Expand the package, which is compressed:
 - a. Open the **Actions** menu.
 - b. Select **Expand**.
 - c. Click **Yes** to confirm.
3. After the package is expanded, turn on **Allow Access**:
 - a. Click the **Conditions** tab and then the **General**.
 - b. Select **Allow Access**.
 - c. Resave the package.
4. Ensure that clients run the automatic installer once to process the package. Use any of the following methods:
 - Email method
 - Logon modification method
 - Central-files method

Customizing the Scheduler

You do not need to customize the scheduler in order for it to work, but you can do so if you wish, customizing it in the following ways:

- ◆ Indicate how long a pause there should be after the scheduler starts and before it begins to process a job.
- ◆ Indicate how often the scheduler should wake and look for jobs to process.
- ◆ Indicate whether users should have a desktop scheduler icon.
- ◆ Indicate whether users should be allowed to terminate the scheduler.
- ◆ Create a message to display to users when a scheduled job is about to run.

These customizations are made at the list level and affect all packages in the list.

▼ To customize the scheduler for a list of packages:

1. Start the console and select **Software Distribution Lists**.
2. Highlight a list in the tree view window and click the **Shared** tab.
3. Click **Runtime** and then **Scheduler**.
4. Modify any or all of the following settings:

a. Pause

The pause indicates how many seconds to pause before beginning a scheduled job. The default is 300 seconds.

The schedule pause and the schedule interval are interdependent. The schedule pause must be at least 3 minutes less than the schedule interval.

b. Interval

The interval indicates how frequently the scheduler agent should check for new jobs to run. The default is 60 minutes.

The schedule pause and the schedule interval are interdependent. The schedule interval must be at least 3 minutes more than the schedule pause. We recommend that you avoid short intervals and make them as much longer than the scheduled pause as is practical. This will avoid a situation where the installer is launched many times because the interval expires before the first install is finished.

- c. Use the **Display Agent Icon...** box to indicate whether you want users to have a scheduler icon on their desktops. By default, they will have the icon.



- d. Indicate whether users should be allowed to terminate scheduled jobs. By default, they are not allowed to do so.
- e. Type a brief message in the **Scheduler Message** box. Users will see this message when a scheduled job is about to be run.

The WinINSTALL NT Service

A Windows NT service is a process which performs a specific system function as a background task without user intervention. Windows NT services are started during system boot, before login takes place.

In WinINSTALL, the automatic installer can run as an NT service. When running as a service, it has the following features:

- ◆ Runs independently of logged-on user
- ◆ Is invisible to user
- ◆ Uses the account and privileges of account under which the server was installed

How the NT Service Locates Resources

The WinINSTALL NT service runs independently of a logged-on user. Therefore, it does not have access to network resources such as shares or drive mappings associated with a particular session. It does have access to network resources via UNC's.

Note You must configure the service to access the appropriate list file using a UNC path if running on a network.

The installer can run from a different location than the list file. The advantage of this is that you can run the scheduler from the local drive and access the installer (and the list file) from the network. On startup, the scheduler does the following:

- ◆ Locates the list file based on the command line. The default is the `Winapps.lst` in the scheduler directory.
- ◆ Locates the installer based on the following search hierarchy:
 - Checks for the installer executable under the App Paths entry
 - Checks the list file directory for the installer executable
 - Checks the `Scheduler.exe` directory for the installer executable

Invisibility to the User

The automatic installer runs in the background and does not present an interface to the user. All communication between the WinINSTALL NT service and the user is through the Windows NT Event Log, accessed via the Event Viewer. Therefore, unless the user is sophisticated or interested enough to check for it, he or she may never know that the service is running and delivering installations to the computer. This can be an advantage for administrators who want to maintain tight control over their company's desktops without being obvious about it.

How NT Service Deals with Accounts

The service runs the installer under the account and privileges that the service was installed with, rather than under the user's account and privileges.

The WinINSTALL NT service account must be set up in the domain through the User Manager. The privileges of the account assigned to the WinINSTALL NT service determine what resources are available to the installer when it is launched by the NT service.

When the NT service is installed to enable client launch, and is later launched with the /ClientLaunch switch, the automatic installer will run using the privileges of the account which was set at the time of the NT service installation, rather than the privileges of the end user who launched the installer.

Deploying the WinINSTALL NT Service

If your users will be using WinINSTALL's NT service, you can prepare their workstations through the NT Service Center.

▼ To deploy the NT service:

1. If you want to deploy the WinINSTALL NT service to Active Directory groups or users, you should install NetBEUI on both the console machine and the target NT server.
2. Before using the WinINSTALL NT Service Center, go to the Windows NT User Manager screen and set up a WinINSTALL service account:
 - a. Give it Domain Administration Equivalent rights.
 - b. Give it a password that does not expire or need to be changed.



When you install the WinINSTALL NT service to all of your workstations, it will use the Domain Administration Equivalent account. It should never get into trouble because of a bad password, and it will be able to access all resources in the domain.

3. After you have created the WinINSTALL service account, start the WinINSTALL Console and expand **Software Distribution Lists**.
4. Open the **File** menu and select **Run**. Then select **NT Service Center**. The NT Service Center will appear with a list of machines on the left and a set of buttons on the right. Icons to the left of each machine in the list indicate whether the WinINSTALL Service Center has been installed and, if so, what its status is:
 - A green traffic light indicates that the service is installed and running.
 - A yellow light indicates that the service, though installed, is paused.
 - A red light indicates that the service, though installed, is stopped.
 - A red circle with a white X indicates that the service is not installed.
5. Select the machines that are to receive the NT Service, and click **Install** in the **Service** area.
6. When the Install Properties screen appears, select the account. If you select **This Account**, give the domain, account, and password.

We recommend using a domain account with sufficient authority on the target machine to ensure successful software installation.

Types of accounts are:

- **System Account**

The system account is a local account, so it does not go through domain logon. As a result, a system account user may not have access to domain network resources.
 - **This Account**

This account can be a local system account or a domain account. A domain account allows the level of authority of the service to be controlled like any other domain user, but requires access to the domain for the service to start. Domain accounts, when used by services, inherit all properties of the domain user, including any network drive mappings.
 - Click the **NetWare Login** button if you want to use a NetWare account.
7. In the **List File** field, indicate what list file the service will use for its list of distribution packages.

8. Set the desired service mode. The NT service can operate in one of three modes:

- Scheduler (default)

The NT service automatically checks for scheduled jobs at the interval defined in the list file.

- Single

The NT service launches the WinINSTALL automatic installer when it starts. The installer is not automatically launched again until the service is restarted.

- Request

The NT service only launches the installer by request. (Either click the **Service Center Launch Installer** button, or include the /Launch switch on the Service Center command line.)

Select the **Enable Client Server Launch** check box if you want selected users to have this option. The end user to whom you have given this option can only launch the installer from the command line, not from interactive mode.

Note When a user launches the NT service by using the /ClientLaunch switch, the installer will run using the privileges of the account which installed it, not those of the end user.

9. If you want to see the current status of the NT service on the selected machines, click **Get Status**. The status will appear at the top of the dialog box.

Note Collecting information on the status of an entire domain may take some time, during which the Service Center could look as though it were not responding.

Intranet Installers

The intranet installers let you launch installations or uninstalls over your intranet. These delivery methods are not actually separate installers. Instead, they provide a special mode of running the automatic installer. For this reason, the end user must have access to the WinINSTALL automatic installer program, `Winstala.exe`.

The intranet installers use a method we call *check point restart*. This means that they pull down all the necessary files to the target system's local disk before initiating the install. If the connection is broken before all of these files can be staged locally, the staging process restarts where it left off. This function is designated for all URL-referenced files (i.e., `<http://something>`) file source (as opposed to UNC path or network share.)



Deploying the Intranet Installer

If you are using Internet Explorer or Netscape Navigator, there are certain steps your users must take, and other steps the administrator must take to enable installations over the intranet.

▼ To deploy the intranet installer:

1. Create a web server if one does not already exist.
2. Modify certain HTM files.
3. Assure that your users have access to the URL given in `Activex.htm` after it is modified.
4. Change, or have your users change, Internet Explorer security and ActiveX settings on any machine where people will want to run the intranet installer.

Creating a Web Server

If a web server does not already exist, you must create one in order to use the WinINSTALL intranet installer.

▼ To create a web server for use with the installer:

1. Start the Internet Service Manager from within the Microsoft Internet Information Server menu option.
2. Expand the tree to show the name of the system that IIS was installed on.
3. Right-click on the system name and select **New** then **Web Site** from the pop-up menu.
4. Complete the New Web Site Wizard:
 - a. Enter the name for the web site and click **Next**.
 - b. Enter the IP address and the port, and click **Next**.
 - c. Browse through Network Neighborhood to find the main WinINSTALL installation directory on the WinINSTALL server.
 - d. Select a user who will access the home directory from this site, and enter a valid password for that user.

- e. Insure that the following choices are selected:
 - Allow Read Access
 - Allow Script Access
 - Allow Execute Access (includes Script Access)

f. Click **Finish**.

There should now be a new web site listed.

5. Right-click on the newly created web site and select **New/Virtual Site**.
6. Complete the New Virtual Directory Wizard.
 - a. Type in an alias.
 - b. Browse through Network Neighborhood to find the main WinINSTALL installation directory on the WinINSTALL server.
 - c. Select the same user that was used for the New Web Site Wizard, and type the password for that user.
 - d. Insure that the following choices are selected:
 - Allow Read Access
 - Allow Script Access
 - Allow Execute Access (includes Script Access)
 - e. Click **Finish**.

There should now be a new virtual site listed.

Prepare the HTM Files for Use

Before modifying any of the files, create a directory called `\HTMORIG` under the WinINSTALL directory on the server and copy `Winstall.htm` and `Activex.htm` to this directory for safekeeping.

▼ To prepare the HTM files:

1. Make safe copies, as mentioned above.
2. Make sure the distribution package is on the web server.



- Using a text editor, modify `Activex.htm` to include the name of the server and the location of both the automatic installer (`Winstala.exe`) and the distribution package you want to process.

The section to be modified is located between `CLASSID` and `</OBJECT>`.

Note All parameter names are case-sensitive. Each must begin with a capital letter; the remaining letters must be lowercase. The entire value is enclosed between quotation marks.

For example:

```
<OBJECT ID= "activex"
CLASSID="clsid:956634E1-1D79-11D1-86B8-00AA00A181F7"
CODEBASE="http://Websitename/Alias/winweb.ocx">
<PARAM NAME="Urlname" VALUE="http://Websitename/Alias/WinZIP.nai">
<PARAM NAME="Path" VALUE="\\WIQA-AAB\Wininstall\WinInstall2000\">
<PARAM NAME="Switch" VALUE="/DeleteNAI">
<PARAM NAME="Temp" VALUE="C:\temp\temp.nai">
</OBJECT>
```

The chart below explains the parameters:

Parameters for the Intranet Installer

Parameter	Function
Urlname	The location and name of the compressed NAI package to distribute. The package must be on a web server. In the example, <i>Alias</i> is the alias that is used to access the virtual directory name of the server where you have put <i>Activex.htm</i> . <i>WinZIP.nai</i> is the name of the distribution package you want to process.
Path	The UNC path to the automatic installer, <code>winstala.exe</code>
Switch	A command line switch for the ActiveX control to pass to the installer. For example, it can instruct the ActiveX control to delete the distribution package once the installation is complete. If this parameter is not specified, the package will not be deleted.
Temp	An <i>existing</i> folder to copy the distribution package to on the user's local system. This path will typically be <code>C:\Temp\Temp.nai</code> . If no path is specified, then the distribution package will be copied to the default download area, which is the Desktop on Win9x and Win NT systems.



Making Winstall.htm the Default Document

After you have edited `Activex.htm`, you must make `Winstall.htm` the default document.

▼ **To set `Winstall.htm` as the default document:**

1. Go to the web server and right-click on the web site name.
2. Select **Properties**.
3. In the website Properties dialog, click the **Document** tab.
4. In the Add Default Document dialog, click **Add** and enter `Winstall.htm` as the default document.
5. Click **OK**.
6. Highlight `Default.htm` and click **Remove**.
7. Highlight `Default.asp` and click **Remove**.
8. When `Winstall.htm` is the only document name in this window, click **OK**.

Changing IE Security and ActiveX Settings

To prepare for using the WinINSTALL intranet installer, your clients must change the security settings for Microsoft Internet Explorer so that they are at medium security or lower. In addition, settings must be modified to enable ActiveX controls and scripts.

▼ **To change security settings and enable ActiveX controls and scripts:**

1. Right-click on the Internet Explorer icon on the desktop.
2. Select **Properties** from the pop-up menu.
3. Once the Properties dialog box is displayed, click the **Security** tab.
4. Click **Sites**.
5. Click **Advanced**.
6. Enter the URL of the Intranet Site that was created.



7. Click **Add** and then **OK**.
8. Click **OK** again.
9. In the **Security Level for this site group** box, click the **Custom Level** button to edit the settings for the site.
10. In the Security Settings dialog, enable all settings having to do with ActiveX controls and scripts.

Configuring WinINSTALL for Netscape Navigator

There are certain steps your users must take and other steps the administrator must take to enable installations over the intranet for users of Netscape Navigator.

On the Users' Machines

First the users must enable access to the browser by clicking the WinINSTALL plug-in to download it. Starting the plug-in will install it and register it with the browser.

▼ **On the user's machine, do the following:**

1. Close the browser.
2. Install the plug-in by copying the `Npchap.dll` files to the plug-in directory. By default this is:

```
Program Files\Netscape\Communicator\Program\Plugins
```

3. Once the plug-in is in the right directory, users restart the browser.

On the Web Server (Running IIS)

▼ **On the web server, do the following:**

1. Stop the web server services.
2. Create a MIME mapping.

To enable the Netscape plug-in, the administrator must create a MIME mapping on the web server for the `.nai` file type. For example, if your web server is running Windows NT, add the required entry to the server's registry:

- a. Start the Registry Editor (`Regedit32.exe`) and navigate to:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
InetInfo\Parameters\MimeMap

Note Note that the "MimeMap" key may not exist. In this case, you need to create the key by right-clicking the **Parameters** key, choosing **New -> Key**, and naming it "MimeMap".

- b. Create a new String value inside the "MimeMap" key by doing the following:
 - Right-click in the right pane of the Regedit window.
 - Choose **New -> String Value**.
 - Give the new string value the NAME (not VALUE):
application/x-chap4,nai,,5
 - Close RegEdit and restart your web server services.

Example HTML Page Source for "Plugin.HTM"

An example of HTML Page Source for `Plugin.HTM` is:

```
<EMBED SRC="http://Web_server/WINSTALL/Somefile.nai"  
PATH="\\WininstallServerName\WINSTALL\">
```

In this example, *web_server* is the name of the server (for example `www.somecompany.com`) where you have put `PLUGIN.HTM`. *Somefile.NAI* is the name of the distribution package you want to install. The `PATH` parameter tells where the automatic installer resides.

If the `PATH` parameter is not included, the plug-in will look for the installer first in the `AppPaths` key, and then in the path.

Email Installer

The email delivery method is actually executed by the automatic installer. The client launches this installer from within the Windows-based electronic mail program by double-clicking a distribution package file that has been sent as a mail attachment. (Some email systems require a method other than double-clicking.)

This delivery method allows the administrator to be very selective about who gets a particular application. In addition, it is a good way to make changes to individual users' Windows configurations without actually having to visit the workstations.



The email installer is compatible with all Windows-based electronic mail packages that support launching applications from attachments, including Microsoft Mail, cc:Mail, DaVinci, Lotus Notes, DEC TeamLinks, and many others.

Email Installer Best Practices

To assure proper functioning of the email installer, the administrator must:

- ◆ Prepare the WinINSTALL console machine to send distribution packages through email.

Most importantly, this just means that the computer must have an email program installed on it. The administrator can simply attach a distribution package file to any email message. Alternatively, with 32-bit email packages, a special **Send** option is available from within the console program. Occasionally, the **Send** option is not active and must be activated.

- ◆ Prepare the users' workstations to use the email installer. To do so, enable the client email distribution package for delivery before the first time `Winstala.exe` is run by the client machines.

Attach the distribution package to an email message and send it to the designated users.

Receiving an Email Application

A user receiving a distribution package in an email message double-clicks the attachment to launch `Winstala.exe`. This program then automatically performs the operation indicated in the distribution package.

The Reset Partition

The reset partition is a hidden area on a client workstation's main hard drive. It holds essential files that let a user reset his or her machine, if necessary, by running the reset workstation application rather than by using a floppy disk. If the user does re-initialize in this fashion, the reset partition will not be removed when the new operating system is installed.

The following files are related to the reset partition feature in WinINSTALL:

- ◆ `WIResAdd.lnk` -- Create Reset Partition link (runs the Reset Partition application)
- ◆ `WIResAdd.exe` -- Create Reset Partition application (puts the partition onto the workstation and copies MS-DOS files to it)
- ◆ `WIResWks.exe` -- Reset Workstation application (reboots from the reset partition to reinitialize the workstation)

- ◆ `WIResRmv.exe -- Remove Reset Partition application (removes the reset partition)`

If the client installs a reset partition, shortcuts are added to the Start Menu under `Programs/VERITAS Software/WinINSTALL`.

The hidden reset partition, once created, can be removed only by running `WIResRmv.Exe` or by resetting from a boot floppy created for an OS template with reset partition disabled.

Deploying a Hidden Reset Partition to Client Machines

There are two ways of setting up the hidden reset partition on a client workstation:

- ◆ Email a Create Reset Partition shortcut file (`WIResAdd.Lnk`), or notify the user where such a shortcut is located. The user then simply runs the shortcut, which in turn runs the application that creates a reset partition on a workstation.
- ◆ Perform an initial workstation reset with a boot floppy created for an OS template that has the reset partition feature turned on.

The first way is generally easier.

Note Every OS template that has the hidden reset partition enabled has a shortcut to the reset program. `WIResAdd.lnk` files are created for each OS template and are unique for each template. Be sure to use the shortcut file specific to the OS template you want to use during a workstation reset.

▼ To find the shortcut file to email to users:

1. Open the console and expand the **OS Distribution Template** branch.
2. Select an OS template for the kind of operating system that should be installed during a workstation reinitialization.
3. Select **Client Operation** in the list view window.
4. Click the Activation Method tab.

The **Install a hidden reset partition** box should be checked on. If it is, the shortcut is listed beneath it.



Running the Shortcut

For Windows NT only, in order to run the Create Reset Partition shortcut from a workstation, the user must have access rights to the shortcut's target distribution site. This means that the user must either log onto the workstation as an authorized domain user, or connect to the distribution site before running the shortcut.

This problem does not occur for Windows 2000 and XP. Those operating systems will request logon to the distribution site automatically.

If a Reset Fails

If a workstation reset fails, then the next reset attempt will may also fail. This happens when a special flag in Cylinder 0 is not removed, causing the reset process to think it has already created the necessary partitions when, in fact, it has not. If this situation occurs, attempt a third reset. This will likely succeed since the special flag gets removed after the second failed attempt.

Customizing the Automatic and Interactive Installers

For NAI packages, you can customize certain ways the automatic and interactive installers behave at runtime. You can do so at either the list level or the package level. You can make the following choices:

- ◆ Enable installer rollback
- ◆ Enable/disable the **Uninstall** button for the interactive installer
- ◆ Disable/Enable the **Abort** button for the automatic installer
- ◆ Set a post-install pause for the automatic installer
- ◆ Set a storage area for application history information for both installers

You cannot use the console to customize MSI packages in this way.

Turning On or Off the Rollback Option

For NAI packages, the installer rollback function is enabled by default. If an installation fails for any reason, WinINSTALL will automatically return all files to the state they were in before you initiated the install procedure.

▼ To set this option:

1. Start the console.
2. Decide whether you are making the setting at the list level or the package level:
 - To make the setting at the list level, highlight a list and select the **Shared** tab.
 - To make the setting at the package level, highlight a package. Select **General** in the list view, and select the **Shared** tab.
3. Click **Runtime** and then **Installer**.
4. Click the check box beside **Recovery Rollback**.

Turning On or Off the Uninstall Option

For NAI packages, you can indicate whether or not users can uninstall any of the applications installed by the interactive installer. The default is to have the remove option enabled.

▼ To set this option:

1. Start the console.
2. Decide whether you are making the setting at the list level or the package level:
 - To make the setting at the list level, highlight a list and select the **Shared** tab.
 - To make the setting at the package level, highlight a package. Select **General** in the list view, and select the **Shared** tab.
3. Click **Runtime** and then **Installer**.
4. Click the check box beside **Enable Interactive Installer Uninstall Button**.

If you are striving for network-wide consistency, you may not want users to have the option to remove applications.



Making the Abort Button Visible or Invisible

You can indicate whether the **Abort** button should be visible and operational to users.

▼ To turn the feature on or off:

1. Start the console.
2. Decide whether you are making the setting at the list level or the package level:
 - To make the setting at the list level, highlight a list and select the **Shared** tab.
 - To make the setting at the package level, highlight a package. Select **General** in the list view, and select the **Shared** tab.
3. Click **Runtime** and then **Installer**.
4. Click the check box beside **Disable Automatic Installer Abort Button**.

Setting the Post-Installation Pause

When the automatic installer has completed processing all packages, it pauses on the user's screen and displays a list of everything it has completed. You can set the pause to a value from 0 (no pause) to 999 (pause until the user clicks the **OK** button). The default value is 30 seconds.

▼ To set the pause:

1. Start the console.
2. Decide whether you are making the setting at the list level or the package level:
 - To make the setting at the list level, highlight a list and select the **Shared** tab.
 - To make the setting at the package level, highlight a package. Select **General** in the list view, and select the **Shared** tab.

Note The usual rules of inheritance do not hold true for this feature. Even if you set a pause at the package level, the package will inherit whatever pause was set at the list level unless you run it from a command line.

3. Click **Runtime** and then **Installer**.
4. Set a number of seconds for the pause.

Setting the Location for Application History

WinINSTALL keeps track of what applications it has installed on a machine. This information is known as the Application History. It is stored in the registry under `HKEY_CURRENT_USER`, `HKEY_LOCAL_MACHINE`, or both. The installers examine the registry (or `Nai.ini`). If the application history shows that an application has already been installed, WinINSTALL will not try to install it again.

▼ To set the storage area:

1. Start the console.
2. Decide whether you are making the setting at the list level or the package level:
 - To make the setting at the list level, highlight a list and select the **Shared tab**.
 - To make the setting at the package level, highlight a package. Select **General** in the list view, and select the **Shared tab**.
3. Click **Runtime** and then **Installer**.
4. Select where to store application history information:
 - Machine Specific (default)

The installer will check `HKEY_Local_Machine` to determine whether to process the package. If the package has been processed on the machine, it will not be reprocessed on that machine, no matter who logs on.
 - User Specific

The installer will check `HKEY_Current_User` to determine whether to process the package. If a new user logs on, the package will be processed again.
 - Both

If you want to accommodate for roving users, select **Both**. The installer will check both `HKEY_Current_User` and `HKEY_Local_Machine`. If the package is not listed in both locations as having been processed, the installers will reprocess the package.

Note This setting does not affect Last Run information. Last Run information is always reported to `HKEY_Current_User`.



Distribution packages are files containing requirements, instructions, and often file sets that the installers use when making the desired changes to your users' PCs. Discover is a standalone program (Disco32.Exe) that creates any of the following types of packages:

- ◆ WinINSTALL Installer Packages

WinINSTALL Packages (NAI) have the file extension .nai. They are created only through Discover and can use the full range of editing functions available through the WinINSTALL Console. They are processed by the WinINSTALL installers.

- ◆ Windows Installer Packages

Windows Installer Packages (MSI) have the file extension .msi. Although these packages can be created by Discover, they are also available from other sources. They can use many editing functions available through the WinINSTALL Console, but they cannot use the full range because MSI packages are very different in structure. They are processed by the Microsoft Windows Installer.

- ◆ Windows Installer Merge Modules

Merge modules have the file extension .msm. Like MSI packages, they contain instructions, components and setup logic. A merge module, however, cannot stand alone. Instead, it must be merged into an existing MSI package. This process alters the original package by adding the merge module's components and logic to it.

- ◆ Windows Installer Transforms

Transform files have the file extension .mst. They are instruction files created by comparing two MSI files and recording the differences between them. These instructions can then be applied as a template to other MSI files to create the same types of differences.



The Discover Program

The Discover program (`Disco32.exe`) is a standalone program that you use to create distribution packages, transforms, and merge modules.

We cannot Discover applications that have been installed by the Windows Installer.

If the Discover client machine is Windows 9x, you can create NAI packages. However, on these earlier versions of Windows, you cannot create an MSI package directly with Discover. The workaround is to create an NAI package and convert it to MSI format by opening the **Actions** menu and selecting **Generate MSI**.

The Discover program works like this:

Step	Process
1	Administrator starts with a clean PC. (See "Best Practices for Using Discover.")
2	Administrator tells Discover to take a picture of a representative PC's software configuration. This is the Before snapshot.
3	Administrator installs or removes an application on the PC that had the Before snapshot taken.
4	Administrator reboots the PC because some application setups are not complete until after the reboot.
5	Administrator runs the newly installed application to verify that it works.
6	Administrator does any fine-tuning that may be necessary to make the application behave as he wants it to. This may or may not necessitate rebooting the system again.
7	Administrator exits the application.
8	Administrator instructs Discover to take an After snapshot of the PC's new configuration.
9	WinINSTALL compares the Before and the After snapshots and notes the changes. It creates a distribution package with instructions on how to install or remove that application on such a PC in the future.
10	(Optional but probable) Administrator uses the WinINSTALL Console to customize the distribution package.

Step	Process
11	Administrator cleans the reference PC to prepare to run Discover again.

You should run the Discover program (`Disco32.exe`) as a standalone executable, outside the WinINSTALL Console Program. When run by itself, the Discover utility does not have a client footprint; nothing is installed on the workstation. This fact lets you take a snapshot of a completely clean PC.

Best Practices for Using Discover

The rules for using Discover are as follows:

- ◆ Run Discover on a clean reference PC, not on the PC where WinINSTALL is installed. A clean PC is defined as a machine with *only* the operating system and any service packs necessary for running the operating system. If you install WinINSTALL or other VERITAS products (or any other products) on the machine, it is by definition no longer a clean machine.

The reference PC should have access to the Discover program (`Disco32.exe`) in the WinINSTALL directory from either Network Neighborhood, Explorer, or the Run line of the **Start** menu.
- ◆ We strongly suggest that you run Discover on a machine using v1.1 of the Windows Installer. Although you can use a greater version, a client with an earlier version will not recognize the package created with the newer version.
- ◆ Do not map a drive to the WinINSTALL share. Discover, which picks up everything, would pick up this specific mapping, and the specificity could cause problems in your distribution package.
- ◆ Create packages on a machine that has the same operating system your users have. Some administrators create images of machines with different operating systems so that the reference PC can be a Windows 98 machine one day, for example, and an NT 4.0 machine the next. Other users have more than one reference PC.
- ◆ Do not create MSI packages using a higher version of MSI than your clients have available. If your clients have MSI version 1.1, they will receive error messages if they try to process packages built for the 1.2 or 2.0 schema. On the other hand, clients with MSI 2.0 or 1.2 can process earlier packages.
- ◆ Create a separate distribution package for each application.



- ◆ Clean the machine between runnings of Discover. That is, return it to a state in which it has only an operating system and any service packs necessary for running the operating system.

Run Discover on a Clean PC

Using a clean PC ensures that the Discover program will pick up all files and registry entries - and only those files and registry entries - necessary for the application to run. The following scenario will help explain the importance of a clean machine.

Assume that you have a DLL called `Abc.dll` on an unclean reference PC. Perhaps you installed it yourself, or perhaps it came from an OEM.

The administrator takes a Before snapshot. Then the administrator installs a new program and takes an After snapshot.

Suppose the new program uses `Abc.dll`. When the new program is installed, it does not install `Abc.dll`, because it is already there. When the After snapshot compares the state of the machine before and after the new program was installed, it does not see `Abc.dll` as a new element, one that had to be installed. Therefore, the instruction to put `Abc.dll` onto the machine would be missing from the distribution package.

Now if a user who does *not* already have `Abc.dll` on their machine runs the distribution package, the DLL will not be installed there, since the instruction was not put into the package. Because the DLL will be missing, the newly-installed application may not work.

Create Packages on the Same Platform Your Users Have

If at all possible, you should run Discover on the same platform as the users' workstations. (For some applications, the service pack level may need to be considered.) Again, this is because your reference PC is serving as a norm. If the users have Windows 2000, a Discover on a Windows 95/98 platform would produce an unpredictable, and possibly non-functional, installation package.

Note If your environment has machines with different operating systems, you should create a package for each operating system.

Clean the Discover Machine Between Runnings of Discover

If you have used a reference machine to install a program, you should clean that machine before creating the next distribution package. Cleaning it means simply to return it to a condition in which it has only the operating system and any necessary service packs.

Because a reference machine need not be large and expensive, some people have two or more. Then they can create a distribution package on one machine while returning another to its clean state. This reduces downtime and therefore reduces the temptation to cut corners by using a machine that is not clean.

Our operating system distribution feature makes it easy to return any machine to a pristine state. Once you have created a floppy for each operating system you might want, you can clean a reference machine simply by booting it up with the appropriate floppy while you are doing something else on another machine.

There are also other commercial packages that let you return PCs to their clean state with very little effort.

What to Do if You Cannot Use a Clean Reference Machine

If you cannot meet the ideal of using a clean reference machine for running Discover, you must use a “quiet” one. Follow these recommendations:

- ◆ Make sure none of the components, particularly any DLLs, of the program you are about to install already exist on the PC.

If a DLL, for example, already exists on the system, the application’s Setup program will not reinstall it and Discover will not record that DLL as part of the installation. When a user tries to start the application, he or she will receive an error message stating that a DLL is missing. Make certain that every element of the installation is accounted for.
- ◆ Shut down antivirus programs.
- ◆ Shut down programs and applications that could write to the PC's temp and/or other directories.
- ◆ Do not use mail and word processing programs.
- ◆ Do not use a browser or have a browser window open.
- ◆ Do not delete files or send files to the recycle bin during the creation of a distribution package.
- ◆ Disable screen saver utilities, email products, personal schedulers, and other programs that run automatically in the background.



Taking the Before Snapshot

If you are not certain what type of package you want to create, see “Advantages of the Different Packaging Formats” on page 55. To create a distribution package, you must first take a Before snapshot.

Note Windows Installer packages can only be discovered on NT systems with the Windows Installer v 1.1 installed.

Starting the Discover Wizard for the Before Snapshot

Start the Discover wizard from the reference machine on which you will be creating the distribution package. This should be a clean machine with nothing on it except the operating system and any necessary service packs for the operating system.

1. Run the Discover program (`Disco32.exe`) from either Network Neighborhood, Explorer, or the Run line of the **Start** menu.

Caution *Do not* map a drive to the WinINSTALL share. Discover, which picks up everything, would pick up this specific mapping, and the specificity could cause problems in your distribution packages.

A wizard overview screen appears. As you complete each screen, click **Next** to continue.

Discover Wizard: Welcome Screen

▼ **To complete this portion of the wizard:**

1. Click **Advanced Options** under either of the following circumstances:
 - If you want to instruct Discover to perform only selected portions of the system scan.
 - If you want to create a transform file (.mst).
2. Click **Next**.

Discover Wizard: Selecting Packaging Format

▼ To complete this portion of the wizard:

1. Indicate whether you want to create a WinINSTALL (NAI) package or a Windows Installer (MSI) package. If you want to create a transform, you must select MSI packaging format.
2. Click **Next**.

Advantages of the Different Packaging Formats

Many administrators know, before running Discover, what kind of package they want to create. Here are some considerations to keep in mind:

- ◆ Some advantages of NAI packages:
 - They have a lighter footprint.
 - They allow you more flexibility in where to put your files.
 - They can deploy a wider range of services and device drivers than can Windows Installer packages.
 - They are not bound to Microsoft's Internet Explorer.
 - They can put source files into a variety of different directories.
 - They use an installer that does not require the Windows Installer to be present on your operating system.
 - They offer more flexible handling of Shortcuts: They provide the option of not uninstalling a shortcut that an .NAI package installed.
 - They let you choose an icon from a .ICO file for your shortcut.
- ◆ Some advantages of MSI packages:
 - They can be deployed via Active Directory group policy objects.
 - They have robust self-repair attributes.
 - They can be self-contained (provided that the PC has the Windows Installer installed on it; all Windows 2000 PCs meet this requirement by default) allowing a user to merely double-click to install the package.
 - They can be edited using software written by other vendors.



Discover Wizard: Selecting Discover Method

This portion of the Wizard is available only if you selected Windows Installer as your packaging format and you checked **Advanced Options** on the Welcome screen.

- ◆ Select **Yes** if you want to create a snapshot that records changes from an application that has already been installed. For example, you should select **Yes** to create:
 - A transform
 - A merge module
- ◆ Select **No** if you want to create a new Windows Installer package (MSI) for an application that has not yet been installed.

Discover Wizard: Specify Installer File

This portion of the wizard is available only if you selected to generate transforms, merge modules, or Windows Installer files that reflect changes made after an application was installed.

▼ To complete this portion of the wizard:

1. Type or browse for the name of the application.

Note Do not use a name that is already being used for a list file.

2. Type any command line parameters to be included in the package.
3. Click **Next**.

Discover Wizard: Advanced MSI

This portion of the wizard is available only if you selected **Advanced Options** on the Welcome screen and are creating a Windows Installer package (MSI). Use it to select advanced attributes which allow more precise control over the NAI-MSI conversion process.

- ◆ Select the MSI schema version for the target MSI package. By default, WinINSTALL will use the latest MSI version that is available on your operating system.

- ◆ Select the **Suppress Advertising...** checkbox to inhibit the advertisement of COM registry entries. The Active Directory data structures which contain the information for Group Policies contain a limitation of 800 advertisement-aware COM servers. When adding a package with a number of advertisement-aware COM server registry entries that exceeds this limitation, the following error message may be displayed:

```
The size of this object exceeds the maximum size set by the Administrator."
```

Selecting the checkbox can circumvent this error condition.

- ◆ If a `mergemodule.xml` file has been created through the console, you can select the **Suppress auto-discovering...** checkbox to inhibit the use of any merge module cache information during the conversion process.
- ◆ If a `mergemodule.xml` has been created through the console, you can select the Update Merge Module Cache. WinINSTALL Merge Module Caching

In WinINSTALL 7.5, a merge module cache can be created for the different merge modules maintained by the WinINSTALL administrator. The cache acts as a quick lookup table for the NAI-to-MSI conversion process for both Discover and the console; the cache contains critical information about the merge module such that when the NAI-to-MSI converter is generating an MSI, if it comes across a package entity that possibly belongs to a part of a merge module, the entire merge module is merged into the MSI being generated as opposed to randomly assigning component GUIDs to that entity. This process effectively emulates a developer making a conscious decision to merge a redistributable into his MSI without the WinINSTALL administrator having to think about it.

Discover Wizard: Selecting Target

▼ To complete this portion of the wizard:

1. Type the name of the distribution package as you want it to appear in the list on the console.
Use a meaningful name. You have up to 40 characters.
2. Click the browse button, and type the name of the distribution package in the File Name window.
 - If you are creating an NAI package, do not give it the same filename as the list file (.lst).



- If you are creating an MSI package (or derivative), you should create a new folder and then open it and create the new package.msi inside this new folder. This is where all Discovered files will go. You will not get the option to select a directory for the Discovered files later as you would when creating an NAI package.
3. Click **Open**.
 4. Select the type of package you want to build:
 - If, earlier in the wizard, you selected WinINSTALL as the package format, there are no more decisions to make on this screen.
 - If, earlier in the wizard, you selected Windows Installer as the package format, you will now choose whether you want to create a regular Windows Installer package (MSI) or a merge module. In either case, a language field displays, defaulted to United States English. Clicking **Browse** brings up a Select Language dialog. Select a language and click **OK**. The language you select will determine the language in which WinINSTALL will display messages during the installation.
 5. Click **Next**.

Discover Wizard: List File Selection

If you are creating a merge module, the List File Selection portion of the wizard will not display.

▼ To complete this portion of the wizard:

1. Type or browse for the name of the list file that will contain the new distribution package.

Note We recommend that you do not use the `winapps.lst` at this point but use a test list instead. This will assure that the package is not accidentally distributed to users before it is ready.

2. If you type the name of a list file that you have not yet created, you will be asked if you want to create it now. Respond with **Yes** to continue or **No** to indicate another list.
If you leave the list file field blank, Discover will create the new distribution package but will not add it to a list file.
3. If you are creating a WinINSTALL package, indicate whether you want WinINSTALL to look automatically for variables to replace. If you do, select the **Scan...** check box.

We recommend that you have WinINSTALL replace server names, etc. with variables. Be aware, however, that this is not a good idea if you have more than one element with the same name, for example, both a server and a directory named Sales.

4. Click **Next**.

Discover Wizard: Specifying Work Drive

Indicate where Discover should store its temporary work files.

▼ To complete this portion of the wizard:

1. Choose a drive to use for storing temporary files:

The Discover process will run much faster if this is a local drive. This drive should have a minimum of 250 megabytes of space available. WinINSTALL will store the files in a temporary working directory, `\Discover.wrk`, and will delete them after the process is completed.

2. Click **Next**.

Discover Wizard: Advanced Options

This portion of the wizard is available only if you selected Advanced Options on the Welcome page. Use it to indicate which items you want Discover to scan. Options you set here are similar to those you could set by using arguments on the `Disco32.exe` command line:

- ◆ Turn off the **File System** box to tell Discover not to find file changes. This is the same as using the `/NOFILES` command line argument.
- ◆ Turn off the **Registry** box to tell Discover not to find Registry changes. This is the same as using the `/NOREG` command line argument.
- ◆ Turn off the **Icons/Shortcuts** box to tell Discover not to find icon changes. This is the same as using the `/NOICONS` command line argument.
- ◆ Turn off the **Text Files** box to tell Discover not to find text file changes. This is the same as using the `/NOTEXTFILES` command line argument.
- ◆ Turn off the **.INI Files** box to tell Discover not to find .INI file changes. This is the same as using the `/NOINIFILES` command line argument.



- ◆ Turn off the **File Attributes** box to tell Discover not to record System, Hidden, or Read-Only attributes. This is the same as using the `./NOFILEATTRIBS` command line argument.

Discover Wizard: Drive Selection

▼ To complete this portion of the wizard:

1. Tell Discover which drive to look on for changes.
 - a. Select at least one drive from the **Available** list. Although you must choose one, it is important to select only the drive or drives where actual changes will occur.

The Windows directory will automatically be scanned for WIN and System INI changes, even if it is not located on the drive selected.
 - b. Click **Add** or **Add All**.

The selections will move to the **Drives to Scan** list.
 - c. If you make a mistake or change your mind, use the **Remove** or **Remove All** buttons.
2. Click **Next**.

Discover Wizard: Exclusion File Selection

▼ To complete this portion of the wizard:

1. Indicate what elements you want to exclude from the present scan. We recommend using the default exclusions.

Note You may not exclude the Windows directory or subdirectories under the Windows directory. Moreover, you cannot browse for a *folder* to exclude in Discover. To select a folder, you must select a file and then, in the edit box, remove the filename to leave just a folder.

If you do want to change the list of exclusions for this run of Discover:

- a. Keep the **Machine's Exclusions** box turned off.
- b. Highlight an item and press delete to remove it, or click the **New** button to add one:



An empty field with a browse button will display at the bottom of the list. Browse for the item you want to exclude.

- c. Repeat for as many items as necessary.

Note Exceptions made with the **Machine's Exclusions** box turned on will *not* affect this particular session, but they will affect later snapshots performed on this same machine. Therefore, you should consider carefully any exclusions you make with **Machine's Exclusions** selected.

2. Click **Next**.

Discover Wizard: Selecting Text Files

▼ To complete this portion of the Wizard:

1. Specify any text files you want Discover to examine.

Discover will automatically examine the Before and After contents of AUTOEXEC.BAT and CONFIG.SYS and will record in the distribution package any changes that it found. It will also examine any text files with the .INI extension. When Discover is looking for added information in any text file, it reads information on a line by line basis. If a line is not marked by a carriage-return, line-feed, or end-of-line character, it reads to the next available end-of-line character and treats all that information as one line.

If you want Discover to examine the contents of other text files and to record any changes, tell it to do so as follows:

- a. Type in or browse for the names of the text files.
- b. If you entered the name manually, click **Add to List**. This step is not necessary if you browsed for the file.
- c. If you mistakenly add a file, highlight it and click **Remove from List**.



- d. Click **Next**.

Note Even if other text files do change, Discover will not record these changes in the distribution package unless you specifically tell it to examine those files. If, after taking the After snapshot, you remember a file that should be edited, use the **Edits** area of the console.

Discover: Exclusion Registry Selection

▼ To complete this portion of the wizard:

1. Indicate what elements you want to exclude from the present scan. We recommend using the default exclusions.

If you do want to change the list of exclusions for this run of Discover:

- a. Keep the **Machine's Exclusions** box turned off.
- b. Highlight an item and press delete to remove it, or click the **New** button to add one:



An empty field with a browse button will display at the bottom of the list. Browse for the item you want to exclude.

- c. Repeat for as many items as necessary.

Note Exceptions made with the **Machine's Exclusions** box turned on will *not* affect this particular session, but they will affect later snapshots performed on this same machine. Therefore, you should consider carefully any exclusions you make with **Machine's Exclusions** selected.

2. Click **Next**.

Choosing Error Suppression

This portion of the wizard is available only if you are creating a transform.

▼ To complete this portion of the Wizard:

1. Indicate which if any errors you want to suppress while the transform is being created.
2. Click **Next**.

Choosing Validation

This portion of the wizard is available only if you are creating a transform.

▼ To complete this portion of the wizard:

1. Check the box if you want to request that the transform and the package have the same language.
2. Click **Next**.

Discover Wizard: Found Files

This portion of the wizard is available only if you are creating a WinINSTALL distribution package.

▼ To complete this portion of the wizard:

1. Indicate how Discover should treat any new files it finds. Click the radio button for one of these three options:

- Do Nothing

Discover will copy the file names to the distribution package with a drive letter of X:\. It will not copy the files to the server. Files will not be accessible for the workstation installation. If you select this option, this will be the last page of the Wizard. Create a New Directory Structure (default)

Discover will create a directory on the server which matches the directory structure created on the reference PC by installation of the new software. It will copy files from the reference PC to the server. Therefore, files will be available for users to complete the installation.



- Accept each directory specification individually
Discover will ask you where you want each directory.

Note If the **Accept ... Individually** option is selected, files needed to complete package processing will be located in many directories and may not be accessible. If the files cannot be found or are not accessible, the job will fail.

2. Click **Next**.

Discover Wizard: Selecting Directory Search

This portion of the wizard is available only if you are creating a WinINSTALL package.

▼ To complete this portion of the wizard:

1. Check the **Search the server...** box if you want Discover to search the server for existing copies of new files it finds.

Note This option is not recommended because the search may take a very long time to complete. Furthermore, enabling this option presents the risk that files needed for other applications will be overwritten with the "new" file found by the Discover process. Using alternate source files can cause copying, rights, program, and installation errors.

If you do select the option, a box appears for you to enter the path to the starting directory for the search.

2. Click **Next**.

Discover Wizard: Specifying Source Directory

▼ To complete this portion of the wizard:

1. Specify the starting directory.

In order for your users to access these files, they must be in a centrally accessible place. Use Universal Naming Conventions.

We strongly recommend the following location:

```
\\Server\Share\Wininstall\UserPkgs\DirName
```

where `DirName` is the same as the name of the distribution package.

2. Click `Finish`.

Discover will begin taking the Before snapshot.

Discover: Completing the Discover Wizard

If you need to make changes, click **Back**. When you click **Finish**, Discover will take the Before snapshot. When it completes, you should install the program you want to install and then restart the Discover wizard to take the After snapshot.

Recommended Naming Conventions

Ideally, you will have installed the WinINSTALL installers to the following area of a server that is accessible to your users:

```
\\Server\Share\Wininstall
```

For ease of management, we strongly recommend that you use the following conventions for storing and naming your files:

1. Store distribution packages, list files, registry files, and Scheduler files in the `WINSTALL` directory, along with the installers. By default, WinINSTALL will put them there.
2. Create a directory under `Wininstall` called something similar to `UserPkgs`.
3. Under the `UserPkgs` directory, create a separate directory for each distribution package, named the same as the distribution package. This would give you a neat and easily-managed set of directories, one for each package. For example:

```
\\Server\Share\WinINSTALL2000\UserPkgs\Word  
\\Server\Share\WinINSTALL2000\UserPkgs\Lotus123  
\\Server\Share\WinINSTALL2000\UserPkgs\FrameMaker
```

Under each package's directory, WinINSTALL would create any subdirectories necessary for installing that package on the users' machines.



Running the Application's Setup Program

After you complete the Before snapshot, install the application exactly as you want it installed by WinINSTALL.

1. Run the application's Setup program to install the application.
2. Reboot the computer.
3. Run the application to verify that it installed correctly.
4. Exit the application.
5. Make any necessary modifications to the install (to toolbars, file storage locations, etc.) now so that they will be included in the After snapshot. Post installation activity, however, should be kept to a minimum. For example, if you register the application or go to a URL, it may interrupt the smooth deployment of the package.

Taking the After Snapshot

▼ **To begin the After snapshot:**

1. Go to the reference machine on which you took the Before snapshot.
2. Run the Discover program (disco32.exe) from either Network Neighborhood, Explorer, or the Run line of the **Start** menu. *Do not* map a drive to the WinINSTALL share.
3. Select **Perform the After Snapshot Now**.

Note We do not recommend using the **Abandon** option. This would start the Discover process over from the beginning and could produce erroneous results because files and Registry entries made by the application just installed are not removed.

Discover will create a distribution package that reflects the installation you performed from the application's Setup program. Discover records any changes to:

- The full directory and file structure of all selected drives and directories
- .INI files on the selected drives
- The system registry
- Your Win.ini and System.ini files

- Your `AutoExec.bat` and `Config.sys` files
- Shortcuts and icon groups

How to Create Distribution Packages Without Discover

For some tasks, you may want to create distribution packages directly through the WinINSTALL, rather than by running Discover. Situations in which you might do this would be:

- ◆ To create a Run Always package
- ◆ To create a System Default distribution package
- ◆ To send a file or small set of files to users, rather than actually installing a product
- ◆ To edit a file or files
- ◆ To place a shortcut onto the Desktop

▼ **To create distribution packages without using Discover, do the following:**

1. Start the WinINSTALL and expand **Software Distribution Lists**.
2. Highlight the list in which you want to place the distribution package.
We recommend a test list. You should not place a new package into a production list until you have tested it.
3. Open the **File** menu and select **New**.
4. Select either **WinINSTALL Package** or **Windows® Installer Package**.
The Create Package dialog appears.
5. Type the path and file name for the new distribution package.
6. Type a description, which is the way you want it to appear in the tree view, and click **OK**.

The new distribution package will appear in the tree view. At this point it is “empty”. That is, it contains no instructions since this was not a file created through the Discover process. You need to put instructions into this empty distribution package. Select an area, such as **Files**, to distribute files, or **Shortcuts**, to create shortcuts, and complete the instructions.



When to Customize Distribution Packages

When you use Discover to generate a distribution package, do not expect the package to work without modification on all machines. Here are some situations when editing is necessary:

- ◆ For security reasons, Discover cannot capture the password used by an NT service. You must edit the distribution package and add the password manually to any services in the list which specify an account. Failure to do so will result in a logon failure for the service. If the NT service uses the local System account, no password is required.
- ◆ The Discover process captures all changes to a machine, intentional or otherwise. In many cases, the simple act of opening a program (for example, Control Panel) results in additions or modifications to the registry. Typically these entries need to be removed manually so they do not create problems on the target machine.

Tip You can avoid this problem by not performing extra tasks between the Before and the After snapshots. Install the application. Make sure it works. But do not open your mail, surf the Internet, etc., between the Before and the After snapshots. Some experienced administrators recommend not leaving the reference PC unattended until the After snapshot has been taken, unless you are certain that no one will use the machine for some purpose in your absence.

- ◆ Some applications are platform-sensitive and install different files for different operating systems. Therefore, Discovering a package on one platform but installing on another can result in an error message. You can work around this problem by creating platform-specific packages, or by scripting the install so that only the correct file is installed.
- ◆ Uninstallation packages can be especially tricky. (See “Avoiding Trouble When Uninstalling Applications.”)

We suggest that you test your distribution packages on a machine that is not part of your production environment.

Note Remember that it is a good idea to examine a distribution package even if you do not need to change it. By doing so, you will learn fine details about the application and how it affects your users' machines.

Problems That Can Occur with Shared DLLs

Occasionally, you may run into a problem where a DLL which is left over from a deleted installation may interfere with a current install. If this situation occurs, you receive an error message. Your choices will be to delete the offending DLL altogether or to rename it so that the WinINSTALL installers do not hang up on the file. Below are two specific instances where this is known to happen.

NwCalls.dll

Problem: If `NwCalls.dll` is present on a workstation and the NetWare Client is not installed, the console program will abnormally terminate because it has loaded `NwCalls` in an effort to request information from the NetWare Client. For example, if you are using the NetWare login name as the key field in an ODBC database, but the NetWare client is not installed.

Solution: Delete or rename the file and reboot the machine. If you want to restore the DLL to its original name, you must choose a different key field first, then rename the DLL.

Winsock.dll

Problem: If `winsock.dll` is present on a workstation and TCP/IP is not installed, trying to use email notification will cause the automatic installer to abnormally terminate. This happens when the automatic installer loads `winsock.dll` to request information about the TCP/IP stack.

Solution: Delete the DLL if it is not needed, or rename it if it is shared by other applications. Reboot your machine.

Avoiding Trouble When Uninstalling Applications

If you have enabled the Uninstall feature, you must be certain that the files, Registry entries, and so forth that will be uninstalled for one application are not being shared by another application. Carefully examine the distribution package to determine if there are shared components. Certain difficulties may arise when applications share components.

One way to avoid problems is to take a Before snapshot of the system with the application on it. Then uninstall the application and take an After snapshot of the system. This should give you a good idea of the system changes the uninstall makes. Examine the changes to make sure that they will not have a negative impact on the rest of your system. If you find something that should not be removed, flag it with the Permanent flag; this prevents subsequent uninstalls that are using the distribution package from removing that element.



Be especially cautious when examining startup files such as AUTOEXEC.BAT and CONFIG.SYS.

Testing a Distribution Package

Once you have completed the Discover procedure, we strongly recommend that you test the newly created distribution package to make sure that it installs and removes the application as you want.

Command Line Arguments for Discover

If you run Discover from the command line, the following arguments are available:

Command Line Arguments

Arguments	Description
/NOFILES	Don't discover any file changes. This restriction can also be set through the wizard's Advanced Options. (See "Discover Wizard: Advanced Options.")
/NOICONS	Don't discover any icon changes. This restriction can also be set through the Wizard's Advanced Options.
/NOREG	Don't discover any registry changes. This restriction can also be set through the Wizard's Advanced Options.
/NOTEXTFILES	Don't discover any text file changes. This restriction can also be set through the Wizard's Advanced Options.
/NOINIFILES	Don't discover any INI file changes. This restriction can also be set through the Wizard's Advanced Options.
/REGONLY	Discover registry changes only.
/FILESONLY	Discover files changes only.
/ICONSONLY	Discover icon changes only.
/TEXTFILESONLY	Discover text file changes only.
/INIFILESONLY	Discover INI file changes only.

The set of flags beginning with `/NO` can be specified cumulatively. That is, you can use any combination of them and in any order. The other flags can be used only one at a time.

Exclusion Files

Discover has a kind of hierarchy of exclusion files:

- ◆ Global scope

At the top of the hierarchy is the `Discover.xml` file, which can be thought of as a global exclusions file for all runs of Discover. Ultimately, for fresh runs of Discover on machines that have never had Discover run on them before, this is the file that tells Discover how to generate the `.xcp` and `.cfg` files in the Windows directory.

- ◆ Machine scope

At the middle of the hierarchy are the `.xcp` and `.cfg` files in the Windows directory, which affect all runs of Discover on the local machine. These files define the Machine exclusion scope. If the files do not exist, Discover creates them by looking in the `Discover.xml` file.

When you run Discover, these files can be accessed by enabling the **Machine exclusions** checkbox in the registry or files exclusion pages. When you click on the **Reset Contents** button while the **Machine Exclusions** checkbox is checked, Discover refreshes the machine scope files by looking in `Discover.xml` for its information.

- ◆ Session scope

At the bottom of the hierarchy are the `.xcp` and `.cfg` files in the `Discover.wrk` directory, which affect the current snapshot session. These three files define the Session exclusion scope. Discover copies these files from the Windows directory prior to performing a snapshot. When you run Discover, these files are modified by not enabling the **Machines exclusions** checkbox in the registry or files exclusion pages. When you click on the **Reset Contents** button while the **Machine Exclusions** checkbox is not checked, Discover copies the `.xcp` file from the Windows directory (Machine scope) into the `Discover.wrk` directory.

If you have run Discover on a machine (and, thus, generated the `.xcp` files in the Windows directory), and you acquire a new `Discover.xml` file, you will need to refresh your machine exclusions to ensure you get the updates; Discover will not automatically do this for you since it cannot know if you want to keep your machine exclusions as they are.

If you have changed your machine exclusions, the changes are not automatically rolled into the session exclusions. You will have to refresh your session exclusions to accomplish that. In this way, you may modify the machine exclusions without having to use those exclusions for the current session.



Tech Note: OS Image and Reset Partition from Boot Floppy

When resetting a workstation with an operating system image and reset partition enabled, the reset partition may be overwritten by the image. This will cause the reset process to fail later.

The problem occurs when the image installer is loading an image onto the entire disk. If the image installer supports “partition loading” - that is, loading an image into a specific partition rather than the entire disk - the reset partition will be preserved.

Partition loading has its downside, however.

- ◆ It is possible that the image install product you use does not have this capability. In this case, you must turn off reset partition creation to successfully reset a workstation.
- ◆ Setting up the image installer’s command-line parameters is more complex and requires that a boot floppy always create a reset partition on a workstation. Also, loading a non-FAT16 image into a partition may prevent the installation of reset partition applications, because the file system will no longer be recognized.

The image installer’s command-line parameters are set in the OS Template’s OS Image property page. Whatever your choice of imaging product, for partition loading you will have to specify, somewhere in the command-line, into which partition the image will be loaded. The partition number you enter will be:

- ◆ The first partition, if no reset partition will be created. Depending on your imaging product, this will be “0” (for zero-based partition numbering) or “1” (for one-based partition numbering).
- ◆ The second partition, if a reset partition will be created (the reset partition is always the first). Depending on your imaging product, this will be “1” (for zero-based partition numbering) or “2” (for one-based partition numbering).

Note The partition numbers you set in the OS Template rely on you knowing in advance if a reset partition will be created by a boot floppy. For OS Templates that do not support a reset partition, this is no problem; you can only create a boot floppy with reset partitioning disabled. However, for OS Templates that do support a reset partition, you must remember to enable the **Create reset partition** checkbox in the Boot Floppy Creation wizard; otherwise, the reset partition will not be created and your image installer will attempt to load to a non-existent partition.

Editing Distribution Packages: The Basics

5

The most important purpose of the WinINSTALL Console is to examine or edit distribution packages.

If you wish to manipulate MSI packages, the Microsoft Windows Installer should be installed on the console machine. This program is an integral part of Windows 2000, but if you have NT 4.0, you must install it yourself. The program for the Microsoft Windows Installer is available in the MSI directory on the CD. Any menu options relating to MSI packages will be grayed out on NT 4 systems until the MSI Installer is installed.



When to Customize Distribution Packages

When you use Discover to generate a distribution package, do not expect the package to work without modification on all machines. Here are some situations when editing is necessary:

- ◆ For security reasons, Discover cannot capture the password used by an NT service. You must edit the distribution package and add the password manually to any services in the list which specify an account. Failure to do so will result in a logon failure for the service. If the NT service uses the local System account, no password is required.
- ◆ The Discover process captures all changes to a machine, intentional or otherwise. In many cases, the simple act of opening a program (for example, Control Panel) results in additions or modifications to the registry. Typically these entries need to be removed manually so they do not create problems on the target machine.

Tip You can avoid this problem by not performing extra tasks between the Before and the After snapshots. Install the application. Make sure it works. But do not open your mail, surf the Internet, etc., between the Before and the After snapshots. Some experienced administrators recommend not leaving the reference PC unattended until the After snapshot has been taken, unless you are certain that no one will use the machine for some purpose in your absence.

- ◆ Some applications are platform-sensitive and install different files for different operating systems. Therefore, Discovering a package on one platform but installing on another can result in an error message. You can work around this problem by creating platform-specific packages, or by scripting the install so that only the correct file is installed.
- ◆ Uninstallation packages can be especially tricky. (See “Avoiding Trouble When Uninstalling Applications.”)

We suggest that you test your distribution packages on a machine that is not part of your production environment.

Note Remember that it is a good idea to examine a distribution package even if you do not need to change it. By doing so, you will learn fine details about the application and how it affects your users' machines.

How to Edit a Distribution Package

No matter how many and how complex the changes you want to make to a distribution package, the basic method remains the same.

▼ **To edit a distribution package:**

1. Start the console and highlight the distribution package you want to work with.

If you are working with an MSI package, you may want to highlight a specific feature or component.

2. Uncompress the package by opening the **Actions** menu and selecting **Expand**.

Tip Make a habit of expanding a package whether or not the package is compressed. Failing to expand a compressed package can cause error messages when you try to save your changes. On the other hand, no problem is caused if you go through the motions of expanding a package that had not been compressed in the first place.

- For an NAI package, respond **Yes** to uncompress the package.
- For an MSI package, give the name and path of the file you want to expand, and then give a different name and path to be used for the resulting expanded file.

3. Navigate to any area you want to edit by selecting categories in the list view and clicking tabs in the data view.
4. Make your changes.
5. Compress the package if you wish.

Note When packages are compressed, the files to copy, Registry files, etc., actually reside within the package; and the installers can access the files directly from the archive, speeding performance. When not compressed, the package only contains pointers to the files.

- NAI packages will be compressed (or not) based on which types of compression you turned on at the list level. To examine the compression options, or to change them at the package level, highlight the package and click **Shared** and then **Compression**. Indicate the compression method to use and the elements to be compressed.
- To compress an MSI package, highlight it, open the **Actions** menu, and select **Compress**.

6. Resave the package. If you forget to save, you will be prompted.



Index

A

- Abort button 46
- ActiveX Settings, changing 39
- Alternative setups 11
 - console 11
 - required rights 12
 - server 12
- Application History 47
- Automatic installer 24
 - customizing 44
 - deploying 25
 - modes 21

B

- Batch file, for client prep 23
- Best practices
 - Discover 51
 - email installer 42

C

- Client installation
 - customize installers 44
 - delivery system, choosing 20
 - deploy automatic installer 25
 - deploy interactive installer 27
 - deploy intranet installer 36
 - deploy NT service 33
 - deploy scheduler 30
 - methods 22
 - reset partition 42, 43
 - system requirements 20
 - through centrally located files 24
 - through email 22
 - through logon script 23
- Components, WinINSTALL 2
- Console 2
 - remote 17

D

- Discover 2

- After snapshot 66
- Before snapshot 54
- best practices 51
- command line arguments 70
- exclusion files 71
- naming conventions 65
- starting directory 64
- using 49
- Discover Wizard
 - Advanced MSI 56
 - Advanced Options 59
 - Choosing Error Suppression 63
 - Choosing Validation 63
 - Drive Selection 60
 - Exclusion File Selection 60
 - Exclusion Registry Selection 62
 - Found Files 63
 - List File Selection 58
 - Selecting Directory Search 64
 - Selecting Discover Method 56
 - Selecting Target 57
 - Selecting Text Files 61
 - Specify Installer File 56
 - Specifying Work Drive 59
 - Welcome screen 54
- Distribution package 3
 - creating 49
 - editing 73
 - testing 70
 - when to customize 68, 74

E

- Editing package
 - basics 75
- Email application
 - receiving 42
- Email installer
 - best practices 42
 - overview 41



-
- Exclusion files 71
 - H**
 - HTM Files, preparing 37
 - I**
 - Installation
 - clients 19
 - console and server 7
 - hardware requirements 9
 - mapping requirements 11
 - software requirements 9
 - system requirements 9
 - system requirements, client 20
 - upgrading 8
 - Installers 3
 - basic, overview 20
 - Interactive Installer
 - deploying 27
 - overview 26
 - Interactive installer
 - customizing 44
 - Intranet installer
 - deploying 36
 - HTM files for 37
 - IE security 39
 - Netscape, configure for 40
 - overview 35
 - web server, create 36
 - Winstall.htm 39
 - L**
 - List files
 - definition 3
 - Logon script
 - NetWare 23
 - Windows 24
 - N**
 - Netscape Navigator, configuring for 40
 - New features 4
 - CD creation wizard 5
 - check point restart 5
 - digital signature support 5
 - enhanced MSI support 4
 - replication error checking 4
 - reset partition 4
 - third-party support 5
 - transforms, support for 4
 - NT Service, WinINSTALL
 - deploying 33
 - overview 32
 - P**
 - Post-installation pause 46
 - R**
 - Reset Partition
 - deploying 43
 - overview 42
 - Rights, required for installation 11
 - Rollback option 45
 - Runtime options
 - abort button 46
 - application history 47
 - post-installation pause 46
 - rollback 45
 - schedule icon 31
 - schedule interval 31
 - schedule pause 31
 - scheduler message 32
 - termination rights 32
 - uninstall rights 45
 - S**
 - Schedule file 3
 - Scheduler
 - customizing 31
 - deploying 30
 - overview 29
 - Shared DLLS
 - potential problems 69
 - System requirements
 - client machines, software 10
 - console, hardware 9
 - console, software 9
 - Discover machine, software 10
 - hardware 9
 - map/share 11
 - server, hardware 9
 - server, software 10
 - software, recommended 9
 - U**
 - Uninstall option 45
 - Upgrading 8
 - User requirements 2
 - W**
 - Web Server, creating 36
 - Wizards
 - Discover 54