

Windows 2000 Server

## Chapter 9 - Improving Software Management in Your IT Infrastructure

---

Using the Software Installation and Maintenance component of IntelliMirror® management technologies in Microsoft® Windows® 2000, you can centrally install and manage software throughout your organization. This Group Policy–based software management system allows information technology (IT) administrators to efficiently install and update software without visiting users' desktops. Software Installation and Maintenance relies on both Active Directory™, the directory service included with Windows 2000, and Group Policy. Therefore, you will find the concepts in this chapter easier to understand if you are familiar with Active Directory and Group Policy.

This chapter will help you understand how to use Software Installation and Maintenance technologies to easily manage the software in your organization. Use the "Improving Software Management in Your IT Infrastructure" worksheet in "Deployment Planning Worksheets" in this book to help you apply the guidelines in this chapter to your unique enterprise IT environment.

### In This Chapter

Overview of Software Installation and Maintenance  
 Software Installation and Maintenance Features and Technologies  
 Deploying Software in Your IT Infrastructure  
 Maintaining Software After Deployment  
 Software Installation and Maintenance Deployment Situations

### Related Information in the Resource Kit

- For more deployment information about Active Directory, see "Designing the Active Directory Structure," in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.
- For more information about Group Policy, see "Group Policy" in the *Microsoft® Distributed Systems Guide*.
- For more information about the Software Installation and Maintenance features of Windows 2000, see "Software Installation and Maintenance" in the *Distributed Systems Guide*.
- For more information about Remote Installation Services, see "Remote OS Installation" in the *Distributed Systems Guide*.
- For more information about using Microsoft® Systems Management Server (SMS), see "Systems Management Server" in the *Distributed Systems Guide*.
- For more information about deploying and installing Microsoft® Windows® 2000 Professional, see "Deploying Windows 2000 Professional" and "Installing Windows 2000 Professional" in the *Microsoft® Windows® 2000 Professional Resource Kit*.

### Overview of Software Installation and Maintenance

One of the most important objectives for organizations today is ensuring that their employees have steady access to the software applications they need to perform their jobs. Equally, organizations with hundreds of thousands of users have come to realize that without the right tools, managing software can quickly become an impossible job. Unintentionally, users can take a desktop environment and quickly turn it into an administrator's nightmare—simply by downloading unsupported software from the Internet, a CD, or other form of removable media, and by deleting files and programs, and more.

The Software Installation and Maintenance component of IntelliMirror enables administrators to create a controlled environment, providing on-demand software installation and automatic repair of applications. The result is that users have steady, trouble-free access to the applications they need to perform their jobs, no matter which computer they are using.

Software Installation and Maintenance enables IT administrators to:

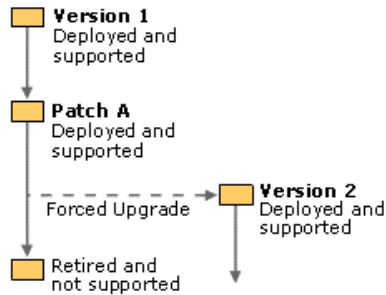
- Centrally deploy new software.
- Ensure that users have the software they need to be productive without an IT administrator or technical support person having to visit each computer.
- Create a standard desktop operating environment that results in uninterrupted user productivity and straightforward administration.
- Maintain version control of software for all desktop computers in the organization.
- Create a controlled environment that can minimize the number of support calls to the Help desk.
- Upgrade applications, deploy patches and operating system upgrades, and remove previously deployed applications that are no longer required.
- Create the rules for how they will deploy, update, and remove software applications.

### Understanding the Software Life Cycle

IT administrators manage software throughout the entire existence of the software in their organizations. This includes deploying, upgrading, patching, and removing software. These phases reflect software life cycle administration. The following is an example of a software life cycle using Software Installation and Maintenance:

1. **Install Version One.** The software life cycle begins when an administrator deploys version one of the software. By using Software Installation and Maintenance, administrators centrally preconfigure and deploy the software for a select group of users, based on their business requirements. Following installation, the users are productively using version one and the administrators are supporting version one.
2. **Install Patch A.** The users who are currently using version one will automatically receive patch A when the administrator deploys it. Because deploying software is a staggered process, administrators support users with only version one and also support those with version one plus the newly installed patch A.
3. **Upgrade to Version Two.** The users who are currently using version one with patch A will automatically receive version two when the administrator deploys it. Using Group Policy, the administrator can configure that all new installations receive version two. Administrators will support users who have version one with patch A and also those with version two.
4. **Retire Version One with Patch A.** After deployment of version two is complete and users have been given adequate notice, administrators can retire version one with patch A.

Figure 9.1 illustrates the example software life cycle described here.



**Figure 9.1 Example of a Software Life Cycle**

Prior to deploying any software into a production environment, the IT administrator must:

- Test the software for potential incompatibility issues.
- Perform a pilot rollout of the software with a group of designated users.
- Provide adequate notice to the users of the software about the eventual removal of the previous version. This is generally done at least one month prior to terminating the software.

### Software Installation and Maintenance Features and Technologies

To take advantage of the Software Installation and Maintenance functionality, you must be running Microsoft® Windows® 2000 Server, with Active Directory and Group Policy on a server, and Windows 2000 Professional on your clients. Additionally, there are other Windows 2000 features and technologies that make it possible for you to take full advantage of Software Installation and Maintenance functionality. While each of the features and technologies are described briefly in the following subsections, they will be discussed in more detail during the course of this chapter.

### Active Directory and Group Policy

In a Windows 2000 Active Directory structure, you can apply Group Policy settings to users or computers on the basis of their membership to sites, domains, or organizational units (OUs). When you create and apply software management–based Group Policy settings to groups of users and groups of computers, you configure settings in a Group Policy object (GPO) that specifies which software is to be deployed, upgraded, or removed from a computer.

By using Group Policy, you can specify software installation–based requirements for your users' environments, and then you can rely on Windows 2000 to continually manage these requirements.

For a complete overview of how your Active Directory structure and Group Policy influence desktop management, see "Understanding How Group Policy Works" in this book. Additionally, for details about how to create effective Group Policy for your organization's user and computer administration needs, see "Developing a Group Policy Implementation Strategy" in this book.

### Software Installation MMC Snap-in

The Software Installation Microsoft Management Console (MMC) snap-in is a resource within Windows 2000 that extends Group Policy and is your primary tool for managing software. By using the Software Installation snap-in, you can establish a Group Policy–based software management system to help you centrally manage:

- The initial deployment of software, including commercial applications and in-house or line-of-business applications.
- The deployment of customized and preconfigured software installations.
- The deployment of software upgrades, patches, and hotfixes.
- The removal of software when users no longer require it or when your organization no longer supports it.

### Assign and Publish

Using the Software Installation snap-in, you can centrally manage the installation of software on a client computer, either by assigning applications to users or computers or by publishing applications for users.

When you assign an application to a user, the next time that user logs on to a desktop the application is advertised to them in the **Start** menu or as a shortcut on the desktop. The application advertisement appears to the user regardless of which physical computer he or she uses, in the case of them having multiple computers. However, the application does not install until the user activates the application on the computer, either by selecting the application from the **Start** menu, clicking on a desktop shortcut, or by activating a document associated with the application. Additionally, the user can go to **Add/Remove Programs** to install assigned applications without having to restart the computer.

When you assign an application to a computer, the installation occurs the next time the computer starts up and the application is available for all the users on that computer.

You can publish software for users only. Those users can choose to install the software from a list of published applications located in **Add/Remove Programs** in Control Panel.

For information about when, why, and how you can assign or publish software, see "Targeting Software" later in this chapter.

### Windows Installer

Windows Installer is a service that helps you manage the state of software during and after installation. It defines a standard format for application setup and tracks components such as groups of files, registry entries, shortcuts, and other aspects of the application that must be managed together. Windows Installer can detect whether software is installed correctly or whether a program file is missing, and can immediately reinstall the damaged or missing files. Additionally, Windows Installer can repair applications and ensure that applications are installed or removed without overwriting or deleting files required by another application. You can also modify the installation by adding or removing features after installation.

The Windows Installer technology uses the following components to install and manage software:

- A Windows Installer package (an .msi file), which is a database containing information that describes the installed state of the application. The Windows Installer package performs the installation, modification, and removal of the software.
- An application programming interface (API) that allows applications and management tools to interact with Windows Installer to install or remove additional features of the application after the initial installation is complete.

**Note** Windows Installer is a base service of the Windows operating system. It is included with Windows 2000 and is available for Microsoft® Windows NT version 4.0, Microsoft® Windows® 98, and Microsoft® Windows® 95.

For more information about installing software with Windows Installer, see "Packaging Software" later in this chapter.

## Distributed File System

Distributed File System (DFS) is a Windows 2000 service that helps you to manage file resources on distributed enterprise networks. You can choose to use DFS to manage the software distribution points in your organization during distribution time. Software distribution points are network locations from which users are able to install the software that they need.

DFS lets you create a path to source files that is independent of the physical location of the source servers. DFS paths are linked into a tree structure, giving you a unified view and means of access to your organization's network resources. For example, \\server1\dataX might appear in the DFS tree as \\Company\Finance\dataX. Users can access the dataX share without requiring any knowledge of which server physically holds the data. It also means that the servers can be changed, managed, or upgraded without adversely affecting users. A single DFS path can map to more than one physical share allowing the load to be balanced between several servers. Optionally, DFS can also replicate data among several servers allowing you to automatically synchronize the data on each DFS replica.

Active Directory enables IT administrators to centrally manage DFS. The DFS tree is published in Active Directory and clients use the Active Directory topology to request the closest replica server to them.

When you use DFS in combination with Software Installation and Maintenance, you benefit from its location independence and the load balancing facilities that can simplify your management and improve and optimize the installation experience for end users.

For more information about using DFS, see "Distributing Software" later in this chapter or "Distributed File System" in the *Distributed Systems Guide*.

## Add/Remove Programs

The **Add/Remove Programs** feature in Control Panel enables end users to manage software on their own computers. IT administrators can control which software is available to users within **Add/Remove Programs** by using Group Policy settings. Users can use **Add/Remove Programs** to install, modify, or remove an existing application or repair a damaged application.

For information about when and why to use **Add/Remove Programs**, see "Targeting Software" later in this chapter.

## Deploying Software in Your IT Infrastructure

IT administrators are constantly managing software in their organizations. With Windows 2000 Software Installation and Maintenance, you perform the following tasks each time you deploy software in your organization whether you are deploying new software or installing upgrades, patches, or service packs:

- **Packaging Software.** *Packaging* software is the process of preparing an application to be installed by a software deployment system.
- **Distributing Software.** *Distributing* software is the process of setting up and managing distribution points or server shares where clients can access the package.
- **Targeting Software.** *Targeting* software is the process of specifying which clients receive the package.
- **Installing Software.** *Installing* software is the process of getting the package onto a computer in a running or ready-to-run state.

You can perform each of these tasks differently with individual software deployment products. Understanding the requirements of each task in the software deployment process for your organization is critical to correctly choosing the appropriate product. The following sections detail how each software deployment task is accomplished using Software Installation and Maintenance.

## Packaging Software

When you package software, you are ensuring that the package contents can be installed appropriately by the installation service. Depending on the application to be deployed, packaging can be as simple as performing an administrative installation to prepare the application for later installation by clients from a network location. Or, packaging can be as complex as repackaging the application, which entails capturing the modifications necessary when installing an application and replacing the application's native setup.

Software Installation and Maintenance uses Windows Installer to deploy software packages. There are a few approaches you can take to package applications for Windows Installer. You can:

- **Use Native Windows Installer Packages.** Many authors of software are developing their applications to make use of the extensive capabilities that Windows Installer provides. These software applications include native Windows Installer (.msi) packages, which are designed modularly to take full advantage of Windows Installer capabilities.  
A native Windows Installer package contains a single product (for example, Microsoft® Office 2000) that is made up of many features that you can configure to install individually. When the user selects an uninstalled feature, the feature automatically installs. This saves bandwidth during the initial installation, it gives the users only the features they need to do their jobs, and it can save space on the user's local disk drive. This is just one of many advantages to using the native Windows Installer packaging format.
- **Reauthor Applications.** If you have mission-critical software applications that have not been designed for Windows Installer and you know they will play an important role in your business in the future, you can reauthor them to include a Windows Installer package. Reauthoring is appropriate for software for which you know what is required to be installed. You will most likely reauthor custom or line-of-business applications because with these types of applications you generally have access to the necessary dynamic-link library (DLL) files, registry keys, and source code you need to reauthor.
- **Repackage Applications.** If you have a software application that has not been designed for Windows Installer, you can *repackage* it into .msi format. This allows you to distribute that software using the features of Windows Installer. A repackaged Windows Installer package contains a single product (and all of the components and applications associated with that product) that is installed as a single feature. Repackaging is most appropriate when you are working with commercial applications, or applications about which you do not have in-depth knowledge.

You probably have many existing applications (in-house applications or other) that do not contain native Windows Installer packages. If you know that your organization will not continue to use these applications in the future, you can simply create software installation settings (.zap) files. If you create .zap files, you will not benefit from the capabilities of Windows Installer but you can still take advantage of some of the features of Software Installation and Maintenance. You can "wrap" the setup.exe or install.exe files into a .zap file format that Software Installation and Maintenance recognizes and you can publish the applications for users in **Add/Remove Programs**.

For more information about assigning and publishing applications, see "Targeting Software" later in this chapter.

## Using Native Windows Installer Packages

A Windows Installer package file is a database that describes how to install a specific product, such as Office 2000. At installation time, the Windows Installer service opens up the package file for the product, and uses the information therein to determine all of the installation operations that must be performed for that product.

With native Windows Installer packages, you can configure for modular installation of components. The modular design enables administrators to configure the installation of the individual application features to occur only when the user selects them. This is called

on-demand or just-in-time installation. For example, you can configure spell checker in Microsoft® Word to install when the user attempts to gain access to it for the first time.

Windows Installer packages enable self-repairing applications, whether they are native or repackaged applications. For example, if a user accidentally deletes Winword.exe, the next time a user attempts to open Word, Windows Installer recognizes that a critical file is missing and reinstalls it automatically.

**Tip** If you plan to develop any new custom or in-house applications, you need to introduce the Windows Installer technology during the planning phase. By developing the application to use the Windows Installer, you can take full advantage of Windows Installer capabilities.

Package authoring tools that developers can use to create native Windows Installer packages are available from a variety of tool vendors. Such authoring tools include, but are not limited to:

- Microsoft® Visual Studio® Installer
- InstallShield for Windows Installer by InstallShield Software Corporation
- Wise for Windows Installer by Wise Solutions, Inc.

It will be easier to manage new software in the future when most software publishers begin to support the Certified for Windows 2000 specification, and will distribute their software with Windows Installer packages and customization capabilities.

For more information about these authoring tools, see the relevant links on the Web Resources page at <http://windows.microsoft.com/windows2000/reskit/webresources>.

### Customizing Windows Installer Packages by Using Transforms

After you have the software in the native Windows Installer package format, you might decide to customize the software for your organization. This is where the modular design of these native Windows Installer packages becomes important. You can customize Windows Installer applications using transforms. Transforms are files you associate with an .msi file at deployment time. You can configure a transform to further customize an installation.

Office 2000 provides the Office 2000 Custom Installation Wizard that lets you build transforms. By using the Office 2000 Custom Installation Wizard, you can create a transform that will enable you to manage the configuration of Office 2000, which you will deploy to users in your organization. You can also use third-party tools to create transforms for Windows Installer packages that do not include their own custom installation tools like Office 2000 does.

You can transform or modify a Windows Installer package to handle a variety of customizations. For example, the same way you can control which features are installed, you can also control where they are located: on the local computer's hard disk drive or on a file server. Although they are optional, transforms can be used for a variety of purposes including the following:

- Encapsulating customizations for particular groups of users to be applied to a base package. This allows, for example, for a single administrative installation point with a base package and separate transforms for the Finance group, Support Staff group, and Legal group.
- Adding new features to an existing application's package. For example, adding custom corporate-developed Microsoft® Excel templates for the Finance group. However, if the templates change frequently, it is a better practice to package them and assign or publish them on their own.
- For customization:
  - Configuring for a set of features from a specified software application or suite of software applications to be installed locally on the computer.
  - Configuring support information links.
  - Preconfiguring mail server parameters (for example, by using Microsoft® Exchange Server).

**Note** It is recommended that transforms are stored at the same software distribution point and in the same shared folders as the Windows Installer package that they customize.

By customizing the Windows Installer package with a transform, you can provide the essential features that users need. In the case of Office 2000, spell checker might be an essential feature, so it installs during the first installation. You might customize less popular features, such as revision capabilities or document translators to install when the user attempts to use them. You might decide that other features, such as clip art, do not install at all or run from the network. Or you can create another transform for Office 2000 that provides all of the features of Word, but what does not install Microsoft® PowerPoint. The exact mix can vary based on the needs of your users.

Associating a transform with an .msi file takes place at deployment time when you are first configuring for the Windows Installer package in the Software Installation snap-in. To associate a transform with a Windows Installer package, select **Advanced published or assigned** in the **Deploy Software** dialog box when you are adding a package to the Software Installation snap-in.

**Important** You must apply the transform before you click OK, at the time of deployment. After you click OK, you can no longer deploy a transform to that deployment. Also, when you create a transform to customize a native Windows Installer package, do not change the product code of the package. If you change the product code, you must verify whether the assignment and publishing of the package with the new product code works as if it were an upgrade or whether it might not be available to users who already have the existing package (with the old product code).

If you need to modify a transform, create a new transform, and then remove the existing application and transform and redeploy the application with the new transform. If you have to do this, consider creating an upgrade relationship between the application and the old transform and the application and the new transform.

For more information about installing transforms, see "Installing Software" later in this chapter.

### Reauthoring Applications

When you reauthor applications you are essentially redeveloping the software to take full advantage of the advanced capabilities of Windows Installer. If you are planning to reauthor an application that does not have a Windows Installer package, you must have the following:

- All the source code, executable files, DLL files, and other resources.
- An understanding of the application and the registry entries, shortcuts, and other information that are needed for it to run correctly.
- A reauthoring tool, such as InstallShield for Windows Installer.

For example, take an organization that developed a custom application that allows employees to arrange for their own business travel. Because the application was developed internally, the organization has all the files for the software, and the developers understand how the software must be installed. This application is ideal for reauthoring for Windows Installer. During the redevelopment process of an application you can repackage it as an interim solution.

You can use the same tools for authoring native Windows Installer packages to reauthor applications.

## Repackaging Windows Installer Packages

Repackaged Windows Installer packages provide the next best software installation and maintenance experience to native Windows Installer packages. Repackaging a Windows Installer package involves taking a snapshot of a clean computer, installing the software, and then taking a post-installation snapshot of the computer. The repackaging software detects the difference of the two snapshots and creates the necessary installation instructions to reproduce the installation.

There are repackaging tools that enable you to repackage software into Windows Installer packages. One such repackaging tool is WinINSTALL LE by VERITAS, which is included with Windows 2000. WinINSTALL LE uses the snapshot technology to build a package. After repackaging, you can use WinINSTALL LE to further customize the package. The aspects of the package you can modify include:

- Required DLL and program files
- Initialization (.ini) file edits
- Registry settings
- General summary information
- Shortcuts
- Services
- Advertised dates, file extensions, programmatic identifier (prog ID), class identifiers (CLSIDs)
- Security settings on files and registry keys

Using WinINSTALL LE, you can make simple modifications to the general summary information of an application that users see when working in the application. It is advisable to modify only the general summary information area unless you are the author of the .msi package. Modifications to the required files, registry settings, and services can make the package unstable, and usually require the assistance of a developer or the original author.

Some additional considerations for repackaging Windows Installer packages include:

- After repackaging, you can use the user interface tool provided in the repackaging tool to inspect for files and registry keys that were both detected by the repackaging tool as having changed between the snapshots and that are not related to the application. Also, during an application installation, pay attention to the extra data that requires you to restart a computer between the before and after snapshots.
- When repackaging an application, the end .msi file provides the same customization as the installation performed between the snapshots. Many applications allow customization during installation. You need to therefore perform the exact customization you want the end user to see. You can also create more than one package for each customization.
- In some applications, you might need to start the application once before the last snapshot because it might install files that need to be part of the package only at the first run.

**Important** Use clean computers when you repackage applications. A clean computer has only the operating system installed and no other software installed or removed. If you do not use a clean computer when you repackage applications, the repackaging tool cannot detect any files or registry entries on the computer that are needed by the application. This can leave vital files out of the repackaged .msi file and might break the application when you try to install it.

For more information about WinINSTALL LE by VERITAS, see the WinINSTALL LE link on the Web Resources page at <http://windows.microsoft.com/windows2000/reskit/webresources>.

## Process for Repackaging Software into Windows Installer Packages

A recommended process for repackaging software into Windows Installer packages is:

1. Start with a clean computer. Consider using the Remote OS Installation component of Windows 2000 to deploy the clean computers in a repackaging lab. For information about using Remote OS Installation, see "Configuring Remote Operating System Installation" in this book.
2. Most repackaging tools require that you take a snapshot of the computer without any applications on it. After you take the snapshot, run the existing installation program for the software you want to repackage, and then configure the files and registry entries for the customizations you want.
3. It is not recommended that the package contain any shortcuts in the startup group. You can either delete any shortcuts placed in the Startup folder by the application between snapshots or manually remove the shortcuts from the Windows Installer package after it is created. In the second case, an application that is assigned to a user activates installations when the user logs on rather than on demand.
4. Delete any shortcuts that point to an uninstall program for the application. A repackaged application must be removed by using **Add/Remove Programs** in Control Panel so that Windows Installer is triggered to perform the removal. The application might not be removed correctly if the user runs the original removal program.
5. When the computer is configured exactly how you want it, take another snapshot. The repackaging tool compares the before and after snapshots and uses the differences to generate a Windows Installer package.
6. Examine the files and registry keys that were detected by the repackaging tool but that you do not want to be part of the package you deploy.
7. Thoroughly test all repackaged Windows Installer packages before deploying them in a production environment. Repackaging is usually a safe process, but some applications that modify system settings can cause problems if they are installed in a repackaged format.

**Note** It is best to repackage applications on the version of the operating system on which they will be installed. This is because the repackaging tool captures the exact image of the application. That image might install a different number of files on an operating system other than the one on which you produced the image.

**Caution** Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or MMC whenever possible.

## Creating Software Installation Settings (.zap) Files

If you have existing applications that you do not want to reauthor or repackage, you can create a .zap file wrapper for the Setup.exe and Install.exe files. A software installation settings (.zap) file is a text file containing instructions about how to publish an application. When you deploy an application using a .zap file, the application installs by using its original Setup.exe file. You can then publish the software so that users can install it by using **Add/Remove Programs** in Control Panel. Additionally, .zap files can provide on-demand installation by file name extension and can initiate an internal Web page when a user selects one to install the application.

**Caution** Consider the following important information when working with .zap files:

- While applications installed using .zap files run their original setup program, they do not run with elevated privileges as Windows Installer packages do. If the application requires administrative privileges to install, users will not be able to install the application unless they are running with administrative privileges
- When Notepad creates .zap files, the files might actually have a .zap.txt file name extension. Make sure that the file name extension of a .zap file is indeed .zap.

The following information is provided as an example for creating a .zap file. Do not copy it exactly. Instead, modify the information to match the application that you are managing and the location of your software distribution point.

The following three lines are all that is needed in a .zap file to publish Microsoft® Excel 97 using the existing Excel 97 setup program:

```
[Application]
FriendlyName = "Microsoft Excel 97"
SetupCommand=" "\\server\share\Excel 97\setup.exe"
```

The path and name of the .exe file must be quoted within the Application section. If there are no command-line arguments, it needs to be quoted twice. For example:

Absolute path:

```
SetupCommand=" "\\server\share\long folder\setup.exe"
SetupCommand=" "\\server\share\long folder\setup.exe" /argument
```

Relative path:

```
SetupCommand=" "setup.exe"
SetupCommand="setup.exe" /argument
```

For more information about creating .zap files, see "Software Installation and Maintenance" in the *Distributed Systems Guide*.

For more information about publishing applications, see "Targeting Software" later in this chapter.

## Distributing Software

After packaging, the application components must be replicated from the location that is initiating the deployment to locations near the intended recipients to provide reasonable performance during installation. This can involve placing and setting up software distribution point servers, replicating the package bits across slow or intermittently connected network links, specifying and coordinating the network bandwidth used for the distribution process, continuously managing which servers need to contain the package, and replicating updates to the package over time.

To begin distributing the software, you need to place the software distribution point file servers, configure network shares for the software, apply the appropriate permissions so that users can access the software, and then replicate the software (including the executable files, the Windows Installer packages, and any transforms) to the software distribution points.

### Placing Software Distribution Point File Servers

When physically placing software distribution point file servers on your network, you must consider the network capacity and bandwidth issues because clients download software across the network and thus increase traffic. You can measure network capacity by determining the number of connections that are established and maintained by the server.

Effective bandwidth varies widely depending upon the transmission capacity of the link, the server configuration, and the server workload. The values for a single server change during operation in response to demand and competition for shared network resources.

With this in mind, you must physically place software distribution point file servers as close as possible (on local networks, whenever possible) to the users who will access them. A client that wants to access a software distribution point will reach the point that is closest to the client within a given site if you are using DFS. If that software distribution point is not available, the directing server will detect and present the client with another software distribution point server within the site. If no software distribution point servers are available within the site, the directing server will present server replicas outside of the primary site.

### Performance Issues Between Clients and Software Distribution Point File Servers

To understand the performance issues between the clients and the software distribution point file servers in your organization, consider analyzing these factors in your test lab:

- Determine the appropriate number of objects that you can include in a Group Policy object (GPO) while taking into account the potential overhead. You need to find the balance between logon times and determine if it is advantageous for you to have more GPOs with fewer settings or fewer GPOs with more settings. Having more GPOs often increases logon times, which can negatively affect users. Having fewer GPOs is easier to manage but provides little flexibility. Testing these configurations in the lab can help you determine which combination is best for your organization.
- The size of each software application that you plan to distribute.
- How long it takes for a particular package to go from point A (software distribution point file server) to point B (client). Test the most common client and server access circumstances, the least common client and server access circumstances, and worst-case situations.

You need to consider how many users access a package at one time. A few instances where large numbers of users might access a package at the same time include:

- **New software.** If the package you are placing on the software distribution point is a new software package, you need to expect a large number of initial installations. This is true whether you publish or assign the software. You might want to stagger the installation so that all users are not downloading packages at the same time on the same day. For example, break up the users into small groups and assign a certain application to 500 users on Monday, assign the same application to 500 different users on Tuesday, and so on. This will minimize the effect that software installation has on your network bandwidth.
- **New employees.** In an enterprise-sized organization, hundreds of new employees might start on the same day. If this happens, consider the network overhead for these users to log on and receive assigned packages. Then consider the number of published packages that these new employees will access over a specified period of time.

You need to test potential issues in lab and determine what might happen on the network if x number of users accessed y software distribution point at the same time on the same day, installing a 10 megabyte (MB) application, a 50 MB application, a 100 MB application, and so on.

### Using DFS to Manage Software Distribution Points

Before you create the software distribution points, you need to determine whether to use the Distributed File System (DFS) service available in Windows 2000. DFS provides a measure of fault tolerance for your software distribution points because it allows a single shared folder to be distributed across several physical servers. In this way, software remains available for installation, even if one of the physical servers on which the software distribution files reside becomes unavailable. In addition, DFS provides server load balancing. Rather than allowing all users to install software from a single server, thus taxing the server, a DFS topology can be designed to distribute network traffic across multiple servers.



You can use DFS to manage the software distribution point servers on your network. Windows 2000 DFS provides two configurations, domain-based and stand-alone.

- Domain-based DFS stores its configuration information in Active Directory. Because this information is available on multiple domain controllers in the domain, domain-based DFS provides high availability for any distributed file system in the domain.
- Stand-alone DFS stores its configuration in the registry of the local computers. It is compatible with earlier versions of DFS.

When you place software distribution points on your network, you can configure one server to become the DFS root, the server responsible for providing referrals for clients to shared folders.

A domain-based DFS root:

- Must be hosted on a Windows 2000–based domain controller.
- Has its topology published automatically to Active Directory.
- Can have root-level shared folders.
- Supports root and file replication through the File Replication Service (FRS).

A stand-alone DFS root:

- Does not use Active Directory (or FRS).
- Cannot have replicas at the root level.

If you have multiple software distribution point file servers in your Windows 2000 domain, any or all participating software distribution points can host and provide fault tolerance for a specific DFS root. Active Directory ensures that domain controllers in the domain share a common DFS topology, thus providing redundancy and fault tolerance.

Installation of the DFS service takes place automatically during Windows 2000 Server Setup. However, you must use the DFS Administrator MMC snap-in to configure a DFS share that is accessible to clients.

For step-by-step instructions about how to configure a DFS share to be accessible to clients, see the Step-by-Step Guide to Distributed File System (DFS) on the Web Resources page at <http://windows.microsoft.com/windows2000/reskit/webresources>.

An alternative to using DFS is to configure multiple distribution points for a Windows Installer package within the .msi file. This capability provides a degree of fault-tolerance for auto-repair and install-on-demand operations.

For information about creating a DFS topology to manage software distribution point file servers, see "Distributed File System" in the *Distributed Systems Guide*.

### Setting up Software Distribution Points

You can begin setting up your software distribution points after you have placed them.

With all deployments, you are required to configure a hard-coded path to the software distribution point. After you have completed the deployment, be aware that you cannot change the path to another software distribution point; that is, without removing the deployment. Removing a deployment can be quite intrusive to users because it can cause the application to be automatically removed from those computers. Using DFS can provide you with the flexibility you might need to make changes to a software distribution point.

For information about using DFS to manage software distribution points, see "Distributed File System" in the *Distributed Systems Guide*.

To begin setting up your software distribution points, copy the software and your Windows Installer packages to the software distribution point file servers.

#### To set up a software distribution point

1. Create the folders for the software on the file server that is designated to be the software distribution point and make the folders network shares. For example: \\<servername>\<sharename>
2. Copy the Windows Installer packages, application executable files, and .zap files to the appropriate shared folders.
 

**Note** Some software supports an administrative setup by using special commands to facilitate the creation of a software distribution point. You can enter the software key once for all users and the server and share name for the software distribution point to which the files are copied.
3. Set permissions on the folders so that only administrators have read and write permissions, and users can only read the files from the software distribution point folders and shares.

Software licenses are required for software that is written by independent software vendors and distributed by software distribution points. It is your responsibility to limit the number of users, who access software through software distribution points, to the number of licenses that you own. It is also your responsibility to verify that you are working within the licensing requirements provided by each independent software vendor.

### Options for Slow Link Connections

Many remote and mobile users connect to the network using slow link connections. Software Installation and Maintenance can move large amounts of data, so processing across a slow link can affect performance dramatically. By default, Software Installation and Maintenance does not operate over slow network or dial-up connections. However, you can change this default behavior by using Group Policy. This way, users can still install published applications over a slow link if necessary, although you cannot configure for automatic assignment, upgrade, or removal of applications to take place over slow links.

**Note** It is recommended you treat any link that is slower than local area network (LAN) speed as a slow link.

For users that are frequently in the office or who generally connect at high speeds, you can rely on the Group Policy slow link detection to apply software settings only when they have sufficient bandwidth. However, if any of your users connect primarily over slow links, you need to define an appropriate software deployment strategy for those users. You can address this challenge in a few different ways:

- To maintain a managed environment, request that portable computer users bring their portable computers in to the office for you to install the software.
- Educate remote users to install applications periodically from **Add/Remove Programs** in Control Panel. Any application that is assigned behaves like a published application over a slow link. It might not be ideal to install applications for remote users, but this way the user can make the decision about when they want to incur the lengthy installation. Keep in mind that if many users share a slow link, they might block the connection. However, if only a few users share a slow link, it might provide a practical solution.
- Publish software by means of a Web server and include that server in the source list of software distribution points for those users.
- Use SMS to deploy software to users who can connect only over slow network links. For more information about providing connectivity for remote and mobile users using SMS, see "Using Systems Management Server to Supplement IntelliMirror" in this book.
- Make it possible for users to be able to connect over a fast link to get their software updates. If these users are able to connect over a fast link sometimes, you can use a script that performs full local installations. The script ensures that the applications are fully

deployed over the fast link so that the user does not receive requests for normally uninstalled (on-demand) components when the user is offline.

### Ensuring Applications are Fully Installed Locally

Use the following script to ensure that applications are completely installed on a local computer. This script performs a full local installation for all applications that are in the advertised state. Note that this script disregards any application that is partially installed. You must therefore make sure that this script runs soon after you advertise the application to the user.

You can deploy this script as a Group Policy logon script. It is recommended to run this script by using the Cscript.exe file. If you run it by using the Wscript.exe file, you might want to remove the wscript.echo statements so that no user intervention is required.

```
' Windows Installer utility to fully install advertised products
' For use with Windows Script Host, CScript.exe or WScript.exe
' Copyright (c) 1999, Microsoft Corporation
'
Option Explicit

'Installstates from msi.h
Const msiInstallStateNotUsed = -7
Const msiInstallStateBadConfig = -6
Const msiInstallStateIncomplete = -5
Const msiInstallStateSourceAbsent = -4
Const msiInstallStateInvalidArg = -2
Const msiInstallStateUnknown = -1
Const msiInstallStateBroken = 0
Const msiInstallStateAdvertised = 1
Const msiInstallStateRemoved = 1
Const msiInstallStateAbsent = 2
Const msiInstallStateLocal = 3
Const msiInstallStateSource = 4
Const msiInstallStateDefault = 5

'UI Levels from msi.h
Const msiInstallUILevelNoChange = 0
Const msiInstallUILevelDefault = 1
Const msiInstallUILevelNone = 2
Const msiInstallUILevelBasic = 3
Const msiInstallUILevelReduced = 4
Const msiInstallUILevelFull = 5
Const msiInstallUILevelProgressOnly = 64

' Connect to Windows Installer object
On Error Resume Next
Dim installer : Set installer = Nothing
Dim product
Dim products
Set installer = Wscript.CreateObject("WindowsInstaller.Installer") : CheckError

'Enumerate through all registered products, checking the installstate
Set products = installer.Products : CheckError
For Each product In products

' Only apps that have nothing installed will pass this check.
If (installer.ProductState(product) = msiInstallStateAdvertised) Then
Wscript.echo installer.ProductInfo( product, "ProductName") & " Is Advertised"
InstallLocal product
Else
Wscript.echo installer.ProductInfo( product, "ProductName") & " Is Installed"
End If
Next
Wscript.Quit 0

Sub InstallLocal( Product )
Wscript.echo "Installing " & installer.ProductInfo( product, "ProductName")

'Set Basic UI, wiht no modal dialogs
installer.UILevel = msiInstallUILevelBasic + msiInstallUILevelProgressOnly: CheckError

'Install the product fully (max feature level)
installer.ConfigureProduct product, 65535, msiInstallStateLocal : CheckError
End Sub

Sub CheckError
Dim message, errRec
If Err = 0 Then Exit Sub
message = Err.Source & " " & Hex(Err) & ": " & Err.Description
If Not installer Is Nothing Then
Set errRec = installer.LastErrorRecord
If Not errRec Is Nothing Then
message = message & vbNewLine & errRec.FormatText
Wscript.Echo message
Exit Sub
End If
End If
Wscript.Echo message
'Wscript.Quit 2
End Sub
```

### Targeting Software

After an application has been packaged and distributed, you need to specify the list of intended recipients. Recipients might be specific users, user groups, computers, or a combination. There might also be other requirements that determine the targeting specifications, such as existing versions of the package or minimum hardware requirements. Special considerations in targeting include the dynamic



nature of any organization and the resulting addition and removal of resources from a target set over time.

You can activate Software Installation and Maintenance by configuring for software installation-based Group Policy object (GPO). Within a GPO you can target software installation by assigning software to users or computers or by publishing software for users.

### Assigning Software

Assign software on a per-user or per-computer basis when you do not want to allow users to choose not to install software or to remove it. For example, if a user accidentally removes a user-assigned application using **Add/Remove Programs**, Software Installation and Maintenance automatically reapplies the advertisement information, and the software is reinstalled the next time a user selects it. It is not possible for a user to delete a computer-assigned application. The following are considerations to observe when assigning software to the users and computers in your organization:

- In most cases, packages that you assign to users or computers need to include applications that are essential, yet will not create excessive congestion between the clients and the software distribution points.
- Determine whether some of the common applications can be included in a Remote Installation Preparation (RIPrep) image. This can reduce lengthy software installation time during the logon process (computer-assigned) or at initial selection of the application (user-assigned). For information about RIPrep and Remote OS Installation, see "Deploying Software Using Remote OS Installation and Software Installation and Maintenance" later in this chapter.
- Assign software to users or computers when you want the software to be resilient. That is, no matter what the user does, the software will always be available.
- If several users use one computer to accomplish their work, and everyone who uses that one computer uses a particular application, that application is an excellent candidate for assignment to the computer.
- Assign mission-critical applications.
- Assign applications that are common among large numbers of users within an organization. When assigning software to large numbers of users, it is most easily done by applying the GPO at the highest level of the Active Directory structure as possible.
- Ascertain the size of each application. Very large applications might not be optimal to install dynamically. For example, a product such as Office 2000 might take a long time to install.

The following is an example of a typical installation process for user-assigned applications:

1. The user logs on to a Windows 2000 Professional-based computer.
2. The WinLogon process advertises applications on the user's desktop or on the **Start** menu.
3. The user invokes the software he or she needs from either the desktop or the **Start** menu, or the user can select a file that has a file name extension for an assigned application that is not yet installed on the local computer.
4. Windows Installer is invoked and it installs the requested Windows Installer package from the distribution point.
5. The shell launches the application.

**Note** If you assign an application to a user, it becomes apparent to the user after the next time the user logs on, which is when the GPO is applied. However, the user can go to **Add/Remove Programs** in Control Panel and install it without logging off and on again.

### Publishing Software

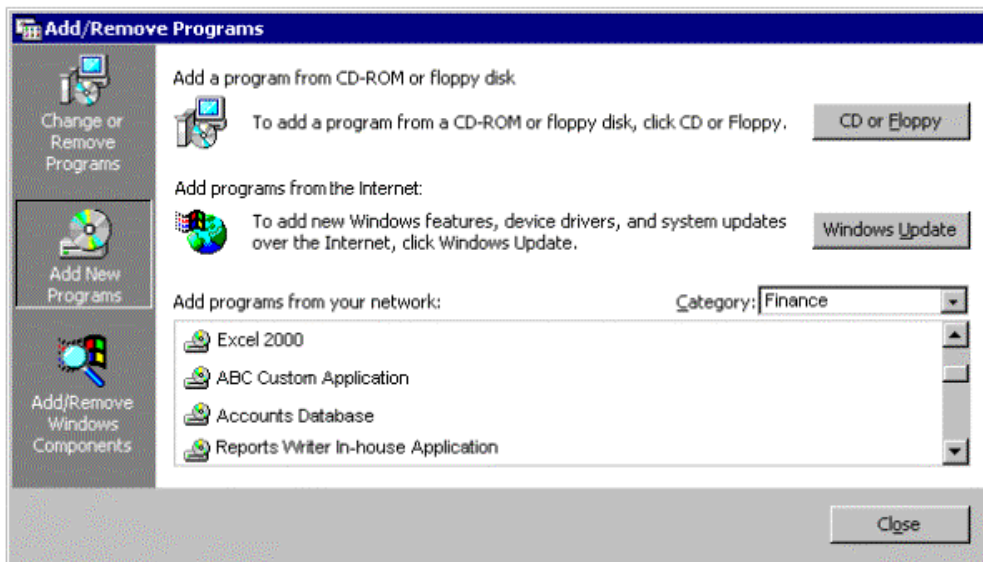
Publish software for users who can optionally install it from a list of published applications in **Add/Remove Programs**. For example, not everyone in the organization requires software for project management. Therefore, an administrator is likely to publish Microsoft® Project for only those users who require it. Then, project managers who require the project management software can elect to install it.

When you publish software for a user, initially it does not appear to be installed on the computer. There is no Windows Installer advertisement information about the software on the computer either in the registry or as shortcuts on the desktop or **Start** menu. The user can install the published software by using **Add/Remove Programs** in Control Panel. Likewise, the user can install the published application when they select a file with a file name extension for an application that is not yet installed on the local computer.

**Note** Because users might be accustomed to installing software from a designated share on your network, it is important that you now educate users about how to install and remove published software by using **Add/Remove Programs** in Control Panel.

Users no longer need to remember server share names and locations. When a user clicks **Add/Remove Programs** and selects **Add New Programs**, a list appears that has available software categories; within these specific categories, the software that is published for them is also listed. Users can install only the software you have published for them.

If there is a large amount of software available, users can quickly locate the software that they need based on the categories that you have predefined. For example, if specified by the administrator, employees in the finance department can only see and install applications in the Everyone and Finance categories. A user from the Finance organizational unit (OU) cannot see or to install applications from the Administration, Sales, or Shipping categories because he or she did not a member of the GPOs from which those applications were deployed. Figure 9.2 provides an example of a list of published applications a user in the finance department might see in **Add/Remove Programs**.

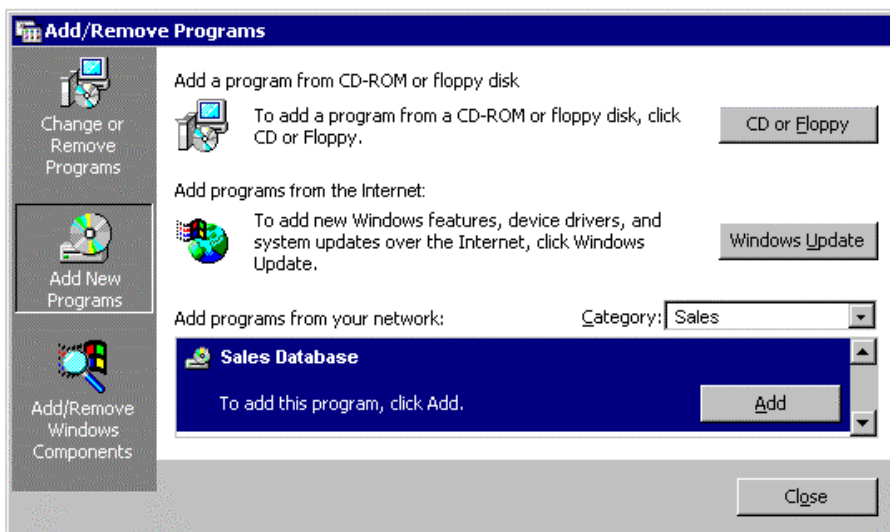


If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.2 List of Published Finance Applications in Add/Remove Programs**

For lightly managed IT environments, you can publish an application to all users at the domain-level without restriction, and then specify a category for the application, such as Sales. In this situation, administrators can expect that users in the sales department will install the software in the Sales category. However, this does not ensure that unauthorized users won't install the software from the Sales category. These are matters you need to consider when publishing software for users in your administrative area. The benefit of publishing software is that it requires less management when change occurs in the Active Directory structure. If you decide to publish an application to smaller groups at lower levels of the domain infrastructure, count on this requiring far more administrative management than to publish to all users at a higher level.

Figure 9.3 provides an example of a published 'Sales Database' application that resides in the Sales category in **Add/Remove Programs**.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.3 Sales Database is a Published Application in the Sales Category Within Add/Remove Programs**

The following are considerations to observe when publishing software for the users in your organization:

- Ascertain the size of each application. Very large applications might not be optimal to install dynamically. A product such as Office 2000 might take long time to install.
- Determine whether you can publish certain applications to all users (without restriction) within your administrative area.
- Create an organized list (perhaps a spreadsheet) of applications that includes the locations from which users can install the application files.
- Determine what to do if the paths to the installation source change. The software installation paths are hard coded in Group Policy, so you need to establish a change management process. However, if you use DFS paths as your installation source path, you can change the physical location of the source, and then reconfigure the DFS path to point to the new location. This is transparent to Group Policy and the users of the application. If an application is packaged as a .zap file, you can only publish it.

The following is a typical installation process for published applications.

1. The user logs on to a Windows 2000 Professional-based computer.
2. The user opens **Add/Remove Programs** in Control Panel.
3. **Add/Remove Programs** provides a list of published software from Active Directory.
4. The user selects the software he or she needs.
5. **Add/Remove Programs** locates the software in Active Directory.

6. Windows Installer is invoked, and then installs the requested Windows Installer package from the software distribution point.
7. The shell starts the application.

**Note** Files with a .zap file name extension can only be published in **Add/Remove Programs**.

After the user installs a published application, it behaves like an assigned application until the user removes the application by using **Add/Remove Programs** or until the administrator removes the application.

### Approaches to Assigning and Publishing Software

Because software can be published, assigned to users, or assigned to computers, you can establish a workable combination to meet your software management goals. Table 9.1 details the different approaches you can use.

**Table 9.1 Approaches to Assigning and Publishing Software**

Situation	Publish	Assign (User)	Assign (Computer)
When the administrator deploys the software, it is available for installation after:	The next time the user logs on. However, it is immediately visible in <b>Add/Remove Programs</b> , if the user is in the right organizational unit.	The next time the user, to whom this application's Group Policy setting applies, logs on.	The next time that the computer is started.
The software is fully installed:	By the user from <b>Add/Remove Programs</b> in Control Panel. Additionally, the application can be installed by opening an associated document if the application is deployed that way.	By the user from the <b>Start</b> menu or a desktop shortcut or by opening an associated document.	Automatically when the computer is started.
If the software is not installed and the user opens a file associated with the software:	The software will install only if Auto-Install is selected.	The software will install.	Does not apply. The software is already installed.
The user wants to remove the software by using <b>Add/Remove Programs</b> :	The user can uninstall it, and subsequently choose to install it again by using <b>Add/Remove Programs</b> .	The user can uninstall it, but the software is still assigned with shortcuts in the <b>Start</b> menu, etc. It is available for installation again from the typical software distribution points.	Only the local administrator and the network administrator can remove the software.

**Note** Windows Installer packages can be assigned to users or computers and can also be published, but .zap files can only be published for users in **Add/Remove Programs**.

The following provides a brief summary of the table categories listed above:

- **Publishing software to users.** After you publish a software package to users in a site, domain, or OU, in most cases the users can go into **Add/Remove Programs** to install the software. An exception exists when you publish an application in a new GPO and must link the GPO to the users in a site, domain, or OU at the same time. In this situation, Software Installation and Maintenance requires the user to either log on to validate the Group Policy change or go to **Add/Remove Programs** and install the application from there.
- **Assigning software to users.** After you assign a software package to users in a site, domain, or OU, the software is advertised on the desktop. The application becomes available in an advertised state to the users the next time that the users to whom this application's Group Policy object applies log on and select it. However, the users can go into **Add/Remove Programs** and install it immediately if the replication took place in replicated domains.
- **Assigning software to computers.** After you assign a software package to computers in a site, domain, or OU, the software installs automatically the next time the computers start up.

**Note** A Windows Installer package must be assigned or published no more than once in the same GPO. For example, if you assign Microsoft Office 2000 to the computers affected by a GPO, do not assign or publish it to users who will log on to the affected computers.

### Categorizing Applications

As mentioned previously, you can organize assigned and published applications into logical categories to make it easier for users to locate the applications that they need in **Add/Remove Programs**.

Windows 2000 does not have predefined categories. You can create and modify categories in the Software Installation snap-in. The categories that you establish are per domain, not per GPO. You need to define them only once for the whole domain. To avoid creating duplicate categories, it is recommended that only one administrator define categories for the entire organization.

If you have a large amount of software to manage, define categories so that users can easily recognize their applications. Create categories that reflect your organizational structure, as when users in a department use a common set of applications. For example, a category named Finance might have Microsoft Excel and other accounting applications. You can also have categories that reflect job functions. For example, a category named Project Managers might include Microsoft Project, Excel, and other project management applications, or you might classify all applications by the type of software. For example, a category named Presentation Tools might include Microsoft® Publisher, PowerPoint, and other graphics programs that are supported by your organization. You can also define broad categories like line-of-business tools, site-licensed applications, and beta applications.

For information about how to create categories see "Categorizing Applications" in this chapter.

### Scaling Group Policy to Meet Your Needs

Group Policy is an extremely flexible tool that you can use to provide users with controlled access to the specific software that they need to do their jobs. You can use Group Policy to control various levels of software administration over users and computers within a specified site, domain, or OU. This section provides examples to help you choose the right Group Policy settings (and other solutions) to deploy and manage software in your organization.

The way you use Group Policy to manage software depends on the network infrastructure, administrative hierarchy, and the business requirements of your organization. Many large organizations can contain a central IT administration with delegated authorization. Some organizations have either strictly centralized or decentralized IT groups. This section covers the scaling of Group Policy to meet the needs of all three types of centralized and decentralized IT organizations.

It is important that software administrators at all levels of the domain hierarchy read the chapters, "Understanding How Group Policy

Works" and "Developing a Group Policy Implementation Strategy" in this book to prepare for creating Group Policy objects to deploy and manage software. Additionally, it is extremely important that you communicate with the other software administrators at all levels of the domain hierarchy to understand which software you will assign, publish, or remove at each level. What software administrators are doing at other levels of the domain hierarchy can affect how you deploy and manage software at your level of the domain hierarchy.

The following situations are examples of where you can scale Group Policy to meet the needs of your organization.

#### Situation 1: Creating a standard operating environment

This situation demonstrates how you can apply Group Policy to assign mandatory, mission-critical software to users and computers by using Software Installation and Maintenance. In this example, you will create two GPOs to assign two software packages—one to all users and one to all computers. You will apply the GPOs at the domain level of the Active Directory structure so that all members in the domain receive the software assignments.

To begin creating a standard desktop configuration, identify the applications that are most appropriate for everyone within the domain, such as virus protection software and e-mail. Depending on your business, there might be custom applications to which all employees must have access.

**Note** In most cases, packages that you assign to users or computers must include applications that are essential yet will not create excessive congestion between the clients and the software distribution points.

It is advised that you assign virus-scanning software to all computers in the organization because this software must function for any user who is operating the computer. Some organizations consider e-mail to be a mission-critical application, but some e-mail packages can be large. Installing large packages over an already congested or slow network link can negatively affect network bandwidth. If this is not an issue for you and you want all users to have e-mail, you can assign the e-mail package to everyone in the organization. After you have configured for software assignment in the appropriate GPO, it is suggested that you apply the GPO associated with these standard software applications to the root domain.

When you apply Group Policy settings high in the domain structure, you can reach more people than if you apply settings at the OU level. You can also decrease application administration of the same application at multiple levels within the domain structure.

#### Notes

- It is currently not possible to inherit Group Policy settings across domains.
- All software administrators within the domain hierarchy must communicate well. If software such as Office 2000 is assigned at the domain level, all users in the child OUs receive that software automatically. This software will likely be managed at the domain level. Managing software at the domain level relieves administrators in the OU levels of the domain from having to plan for configuring for the software installation of certain software at their level.

Figure 9.4 illustrates how you can assign software to users and computers by applying application-based GPOs at the domain level and how that software assignment is inherited by the associated OUs by default.

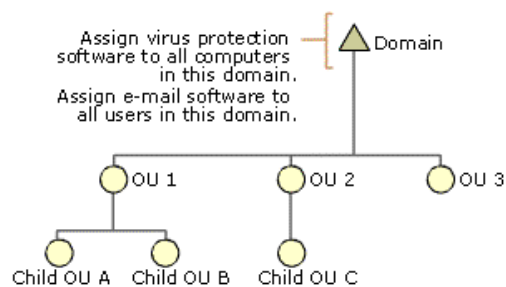


Figure 9.4 Assigning Application -Based GPOs at the Domain Level

#### Situation 2: Publishing software for users

This situation demonstrates how to publish elective, nonessential software for users in **Add/Remove Programs**. For example, your organization might have an in-house application, such as a corporate organizational chart, an employee locator application, or another type of custom software. Published applications are usually nonessential to the user for performing his or her job. If a user chooses to access these applications, they can select them from **Add/Remove Programs** in Control Panel, and then Windows Installer installs the applications. You can publish these types of applications for all users or only a select group of users. When you create the software installation-based GPO you will apply it to the specified site, domain, or OU.

#### Situation 3: Provide users who have common software requirements, and who are spread out across multiple OUs, access to software by using security group filtering

This situation demonstrates how to assign or publish software to users who require access to similar business applications but who are located in different OUs within the same domain. You can refine which groups of computers and users within a site, domain, or OU that a particular GPO influences by using security groups. Security groups are groups that you can create to administer permissions for users and other domain objects within a domain infrastructure.

Consider an organization with an Active Directory structure where the sales personnel user accounts are distributed among four organizational units named for their sales offices: Seattle, San Francisco, Boston, and New York. The organization does not want to apply identical software-based Group Policy settings to all the users in each OU. While all members of each OU might share access to common applications, it is equally possible that sales representatives, telemarketers, and vice presidents will require software that clerical workers who reside in the same organizational unit do not. Security and client administrators can organize users and computers into security groups, and then use these groups to filter the impact of Group Policy.

Microsoft produces multiple versions of Office 2000. It is possible that clerical workers within an organization's domain hierarchy might only require the Standard Edition of Office 2000, while other knowledge workers require the additional functionality that the Professional Edition of Office 2000 offers. To provide all users with the software that best meets their job requirements, you can break these users into separate security groups to filter their access to applications, and then apply GPOs that assign or publish the applicable software to each group of users.

**Important** In most cases, whether you create a new OU depends on factors outside the scope of Software Installation and Maintenance. With regard to Software Installation and Maintenance, use security group filtering only if the software requirements for users within the same site, domain, or organizational unit differ minimally. If more complex software requirement differences exist among the users within the same site, domain, or OU, it might be better to create separate OUs.

For information about creating and administering security groups, see the *Distributed Systems Guide*.

#### Targeting Software to Multilingual Users

Multinational corporations share information and collaborate on a worldwide basis. Up to this point, the operating system support and the files that you generated by using an application in one language were not necessarily compatible with a file that was created by using the same application in a different language.

Microsoft offers several versions of the Windows 2000 operating system to meet the needs of multinational and multilingual businesses working with the challenge of language barriers. All the versions provide sophisticated multilingual functionality, such as the ability to edit and view content in multiple languages. In the English version of Windows 2000, the user interface elements, such as menus, dialog boxes and Help files, appear in English. For businesses that need the Windows 2000 user interface in a language other than English, Microsoft offers Windows 2000 Professional in 23 other localized languages. For businesses that require the Windows 2000 user interface in multiple languages but do not require an entirely localized UI, Microsoft offers Windows 2000 MultiLanguage. In this version, approximately 95 percent of the operating system is localized, allowing users to switch between multiple languages at any time, while the rest of the operating system remains in English.

Organizations using Microsoft® Windows® 2000 MultiLanguage Version can roll out one operating system worldwide rather than deploy a localized version of the system in each country where they operate. For example, a United States–based corporation with subsidiary offices in Tokyo, Athens, and Rome can now deploy Windows 2000 MultiLanguage Version in each office rather than deploying the Windows 2000 Japanese version in Tokyo, the Windows 2000 Greek version in Athens, and the Windows 2000 Italian version in Rome. The time and expense of system administration is significantly reduced by the use of one system worldwide. Rather than waiting for and then applying, tracking, and maintaining a different service pack and tool and application set for each localized version, IT administrators can apply the same software to every computer in the company, regardless of location.

Windows 2000 MultiLanguage Version eases system support and administration. Support personnel can set the workstation to English or any other UI language to more easily troubleshoot issues, and then reset the language when the issues have been resolved. System administrators and support personnel can set the language of the workstation to English or their own preferred user interface language to more easily configure the computer for users. After changes have been made, the user interface language can be reset to the language of the computer's primary users.

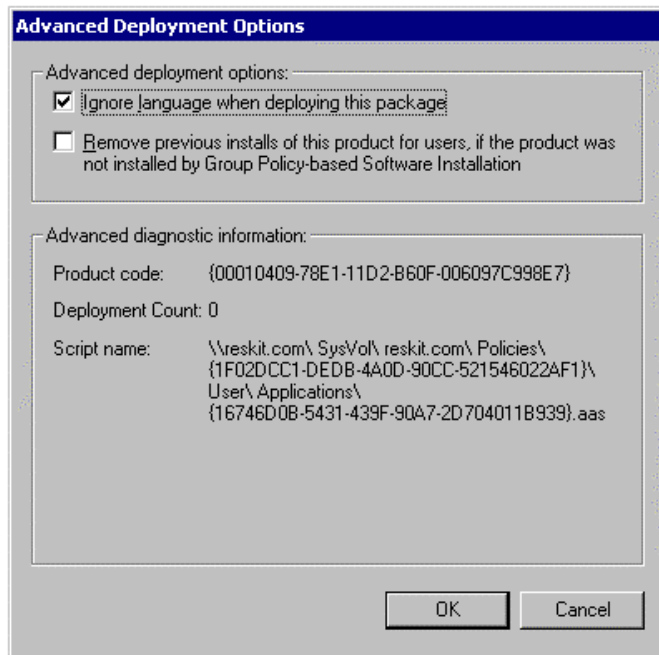
Finally, Windows 2000 MultiLanguage Version allows workstations to be shared by several users with different language preferences, eliminating the need for dual-boot configurations or dedicated computers for each language. This functionality helps workers at multinational organizations be more productive by providing them with a friendly system environment.

Although a Windows 2000 Professional–based computer has many different locale and language settings, for the purposes of Software Installation and Maintenance, only the system locale matters. The system locale determines the code page that is installed and used by non-Unicode applications. Code page availability, more than any other factor, affects the behavior of a language-specific or multilingual application. For additional information about locale and language settings, see the Locales and Language link on the Web Resources page at <http://windows.microsoft.com/windows2000/reskit/webresources>.

Applications typically support one language. This supported language, or the product language, is expressed in the Windows Installer package for the application.

To determine whether to install an application on a Windows 2000–based computer, Software Installation and Maintenance verifies the following:

- If the system locale and the product language match, the package is either installed or advertised.
- If the system locale is any value and the product language is either neutral or English, the package is either installed or advertised.
- If the Ignore language parameter is set in the Software Installation snap-in for the managed package, the package is either advertised or installed whether or not the system locale and product language match. Figure 9.5 shows that this option is set.



**Figure 9.5 Ignore Language When Deploying This Package**

**Caution** Be careful when deploying two versions of an application with different product languages, such as an English and a German version of the same application, in the same GPO if the two applications have the same product code. If they have the same product code, only one of them is installed for users or computers. If you need to make the same application available to users in multiple languages, the different language versions of the application must have different product codes.

### Conducting a Pilot for Software Installation

Software deployment affects both client and server performance. Within a pilot plan, consider strategies for optimizing software deployment on your network and on clients. You must clearly understand the current arrangement of servers, clients, and users throughout your network before you attempt to deploy any applications. For example, you need to consider the impact on the network performance of allowing software to be downloaded over slow network connections. You must also carefully define who can install and update applications. These and other considerations greatly affect the number of support calls that you receive.

It is recommended that each application you plan to deploy be evaluated from the perspective of the users who require the application. When creating a pilot plan for software installation, consider the following:

- Whether everyone in the company requires access to the software. If not, identify the select group that requires access.

- Whether to assign to users, assign to computers, or publish to users.
- If there are any remote software installation requirements.
- How often the software requires updates.
- If you need to customize the software for any divisions in the company.
- How often you need to update user templates or custom files that you want to manage by using the software.

After making these determinations and getting to know your users and their software needs, you can establish a pilot plan, conduct the pilot, and manage the software.

During the pilot, you need to experiment with all conceivable methods for deploying software. If you want to distribute additional templates or files with an application to help just the Help desk personnel, place the templates and files in a separate Windows Installer package. Then assign or publish both the application and the templates package to the appropriate OU. When there are new templates, files, or changes to these files, you can upgrade the existing templates package. You do not have to reinstall the entire application. If you add the templates to the application package during the initial deployment, the entire application needs to be upgraded when a new template or file is required.

#### Deploying Software: Software Installation User Examples

In "Assessing Organization and User Needs" in this book, you learned to categorize user types by job function and by location. Although this section does not include examples of all the user types presented in that chapter, it does provide you with information about some of the most common types of users in organizations around the world.

The following examples explore some of the issues that can arise when deploying software. These examples can help you during your pilot because they highlight some best practice tips that Windows 2000 Software Installation and Maintenance supports.

Although it is hard to provide a single solution for all organizations and all situations, these examples illustrate some commonly encountered situations. They include recommendations about how best to use Software Installation and Maintenance for specific types of users.

#### Stationary Workers

In most organizations, each worker has their own computer and they do not move. They work from the same office each day. Typically the computers they use are desktop computers, connected to the network with a consistent, high-speed connection.

Administrators might want to make these clients their base model, and define a Standard Operating Environment (SOE) that includes standards on how to install and configure the operating system, how to use Group Policy to further configure and manage the environment, and which software is needed.

If these computers have the necessary hardware to take advantage of the Remote OS Installation feature of Windows 2000, the administrator can include the necessary application software with the operating system image. The software is then copied onto the computer with the operating system. For more information about this, see "Bringing Unmanaged Applications into a Managed State" and "Deploying Software Using Remote OS Installation and Software Installation and Maintenance" later in this chapter.

If these computers do not have the necessary hardware to take advantage of Remote OS Installation, the administrator can use the Software Installation snap-in to assign software to the computers. They do this by editing the GPO that manages the computers, and by using the Software Installation snap-in in the Computer Configuration of the Group Policy namespace to assign software to the computers.

#### Roaming Users

In many organizations, some users move or "roam" from one location to another. For example, some receptionists are considered roaming users. It is important to understand that although the roaming user logs on to different computers to perform their jobs, these computers are usually connected by a high-speed connection or a LAN connection. Additionally, managed applications follow the roaming user no matter to which computer he or she is logged on. For example, when a user logs on to computer A and installs software, and then goes to another location and logs on to computer B, the user sees that the software that is installed on computer A is also installed on computer B. Remember that any modifications made for or by the user on one computer are propagated to all other computers the user logs on to in the future.

**Note** Because Software Installation and Maintenance Group Policy settings are not applied over a slow link by default, roaming users who connect to their organization's network by slow links might not see changes to their software. You can change this default behavior in Group Policy so that applications are installed over slow links. You can also change the network speed threshold that Group Policy uses to decide whether a link is considered "slow." For more information about slow links and Group Policy, see "Understanding How Group Policy Works" in this book.

When configuring for roaming users, you can assign applications to either the users or to the computers. For example, if all receptionists use a certain application, it makes sense to assign that application to the computers so that it is already installed on the computers that the receptionists use. Sometimes you might need to assign the software to the receptionists as users. Then, when receptionists relieve one another, they each can log on to the computer and see the applications assigned to him or her because the application is advertised. The application is only installed for the user if they actually run the application.

#### Mobile Users

Some users travel extensively to perform their jobs. For example, sales personnel often spend more time at the offices of customers than at their own. Remember that although mobile users log on to the same computer, their computers sometimes connect through a high-speed line and sometimes through a low-speed (or dial-up) line, and that some mobile users never have a fast connection.

One challenge in assigning applications to mobile users is that source files might be unavailable.

#### Notes

- When connected over a slow link, user assignment effectively behaves the same as publishing software to users. Assuming that Group Policy slow-link processing is set to Default in the user interface, the software does not install on demand. Users can go to Add/Remove Programs to install the assigned software there.
- If users experience difficulty staying connected when they download software, verify that the connection speed and Group Policy settings are set appropriately. When using Windows 2000, you can define the connection speed that is considered to be a slow link.
- For this type of user, it might be necessary for you to configure it so that all important software components defined by you or the user are installed initially. This allows the user access to the necessary software components even when he or she is not connected to the network.

#### Sharing Computers

In many organizations, users share computers. This is quite common on the factory floor or in a business such as a bank, where tellers have the same job function but might work at a different counter and computer on different shifts. In these environments, the software is often task-based. Although users change, the software does not; nevertheless, the software might track who is logged on. You can group these users or computers to manage them from a single GPO, and then use Software Installation and Maintenance to assign software to the computers either by the **Computer Configuration** or the **User Configuration** of the GPO namespace. The software is then available for every user of that computer. You can use this feature to manage the software and essentially reinstall the software environment every time the computer restarts.



Be aware that when new software is assigned to a computer, it is installed when the computer restarts. If computers restart between shifts, the new software installation or upgrade might affect the total startup time of the computer. Remember that this increase in startup time occurs only if new software was assigned or the existing software is upgraded. Another place where computers are shared is a computer lab or classroom environment where users typically share computers for a short period of time, such as while they are taking a class. This scenario differs from the previous shared computer scenario because each user might use the same software as the previous user (for example, because they are in the same class) or they might use different software. These computers likely do not move. Therefore, in this case using a site to manage software makes sense, although grouping the computers into a single organizational unit can also work. Choose the method that gives you the right level of control for applying Group Policy and Software Installation and Maintenance. Depending on the requirements, you might decide to assign software to the computer. This can work well if the software is written to keep separate user information such as configuration information and saved files from software information such as executable files. Another way to manage this environment is to assign software to the user so that each user gets the software that they need for their training.

Group Policy can also limit the changes that users can make to the desktop and other settings on these computers so that you can tightly manage each computer.

**Note** Consider using Remote OS Installation in these shared computer environments. That way, if you ever have to rebuild the shared environment, you can do so in a quick and efficient manner.

## Installing Software

After a specific user or computer receives the targeting information and package details, the installation process must occur to complete deployment. Installation involves specifying exactly when and how the computer settings, the user settings, or both settings must be modified to use the application. The following are some of the many installation issues:

- Scheduling the correct time, event, or system state before the installation begins.
- Coordinating the permissions and access rights necessary to perform the installation.
- Upgrading or removing existing package versions.
- Installing the package for all system users or for individual users of a system.
- Copying the package locally or running it from a network location.

Software Installation and Maintenance provides a range of installation and management configuration options that you can define by using Group Policy. A GPO can contain several settings that affect how an application is installed, managed, and removed. By using the Software Installation snap-in, you can configure the software installation options for each Windows Installer package.

When you begin configuring for the installation of software in your enterprise, you need to understand when, why, and how you can use each option. This section will familiarize you with these configuration options. By understanding the options, you can plan for the impact each configuration might have on the users in your organization.

Before you begin configuring for the installation of software by using Software Installation and Maintenance, you must obtain a Windows Installer package and place the installation files on the software distribution point file server. When you initiate the Software Installation snap-in, you can begin configuring the options in the user interface.

The following sections show the most significant configuration options available in the Software Installation snap-in user interface, and provide the information for when, why, and how you can use each configuration option.

### Configuring the Application Properties

You can manage several settings for each application in the application's **Properties** dialog box, located in the Software Installation snap-in. This section outlines these application deployment properties in the Software Installation snap-in.

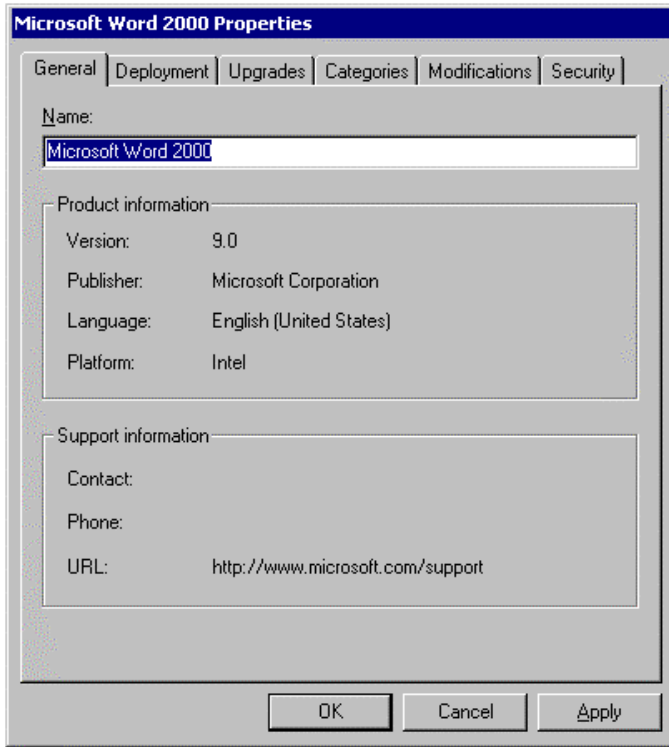
To begin the tour of the application properties dialog box:

1. Choose a GPO for a specific site, domain, or OU.
2. Open the Group Policy snap-in, and then open the Software Installation snap-in.
3. Right-click the software installation node, then click **Properties**.

For more information about how to navigate to the Group Policy snap-in, see "Developing a Group Policy Implementation Strategy" in this book.

### General Tab

Figure 9.6 shows the **General** tab for Microsoft Word 2000 properties sheet in the Software Installation snap-in.



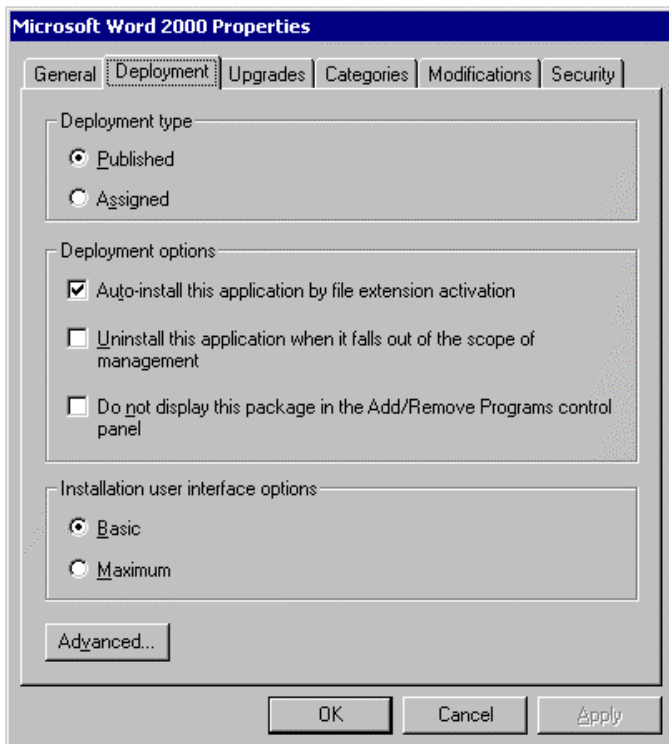
If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.6 General Information About The Application**

The **General** tab is an excellent point of reference for application-specific and support information.

**Deployment Tab**

Figure 9.7 shows the **Deployment** tab for Microsoft Word 2000 properties sheet in the Software Installation snap-in.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.7 Application Deployment Properties**

Table 9.2 describes the options that are available from the **Deployment** tab of an application's properties sheet.

**Table 9.2 Descriptions of Application Deployment Properties**

Areas	Options	Functions
Deployment type	Published Assigned	Publishes the application for the users. Assigns the application to the users or computers.

Deployment options	Auto-install this application by file name extension activation	Automatically installs a published application when a user double-clicks an associated file name extension and the software associated with that extension is not already installed.
Deployment options	Uninstall this application when it falls out of the scope of management	When the GPO that deployed an application no longer applies to a particular user or group of users, the application is removed.
Deployment options	Do not display this package in <b>Add/Remove Programs</b> in Control Panel	Prevents the application from appearing in <b>Add/Remove Programs</b> . If a user double-clicks an associated file name extension or gains access to an advertised shortcut, the application is installed.
Installation user interface options	Basic Maximum	Determines how much information the user sees during installation of the application: <b>Basic</b> provides only error messages and progress bars. <b>Maximum</b> shows the user any information about the installation that the creator of the package provides.
<b>Advanced</b> button	Ignore Language Remove previous installations of this product Advanced diagnostic information	Ignores language when assigning or publishing a package. If an existing unmanaged version of the application is already installed on the computer, removes previous unmanaged installations of the product, and then replaces with a managed version of the software. Displays diagnostic information including product code, deployment count, and script name.

**Notes**

- Use the **Uninstall this application when it falls out of the scope of management** option to avoid the existence of unmanaged, orphaned applications. If some applications have security sensitive information, ensure that users can no longer use the applications after you stop supporting them.
- Use the **Remove previous installations of this product** option to avoid potential security issues. When you bring an unmanaged application into management, someone might gain access to the system and deploy malicious code. Setting this option can protect against that possibility.

For step-by-step procedures about configuring deployment options, see Windows 2000 Help.

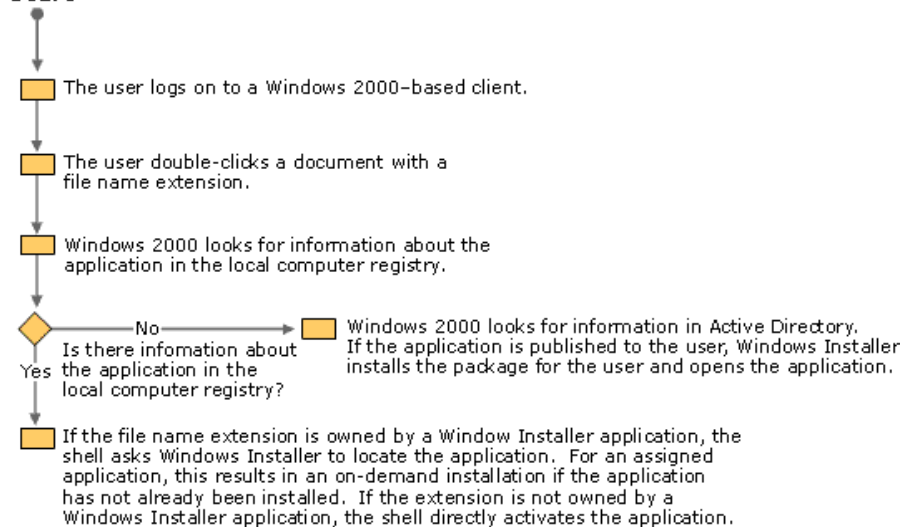
**Auto-Install**

On the **Deployment** tab as displayed in Figure 9.7 you can choose to automatically install an application by file name extension activation. Without regard to whether an application is assigned or published, users can force an application to be installed by invoking a document or file that is associated with the application.

Users often receive documents as attachments in e-mail messages without having the software associated with the attachment installed on their computer. When given the options, in most cases users do not know which file type to choose to successfully open the attachment. With Windows 2000 Software Installation and Maintenance, you can configure the appropriate software to install automatically.

Figure 9.8 shows the process of software being installed by document invocation.

**Start**



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.8 Process For Installing Software by Document Invocation**

A user double-clicks an unknown document or attachment to open it. If an assigned application is advertised on the user's computer but the application is not yet installed, Windows Installer installs the application and opens the document.

If no application, either installed or advertised, exists on the computer for that document type, Windows 2000 Professional looks in

Active Directory for the published application that is associated with the attachment file name extension.

If the application is published, Windows Installer uses the advertisement script for the application in Active Directory to install the application and then opens the document for the user.

**Important** If you have more than one application assigned to the same file name extension, the last one installed on the local computer is the application that opens the file.

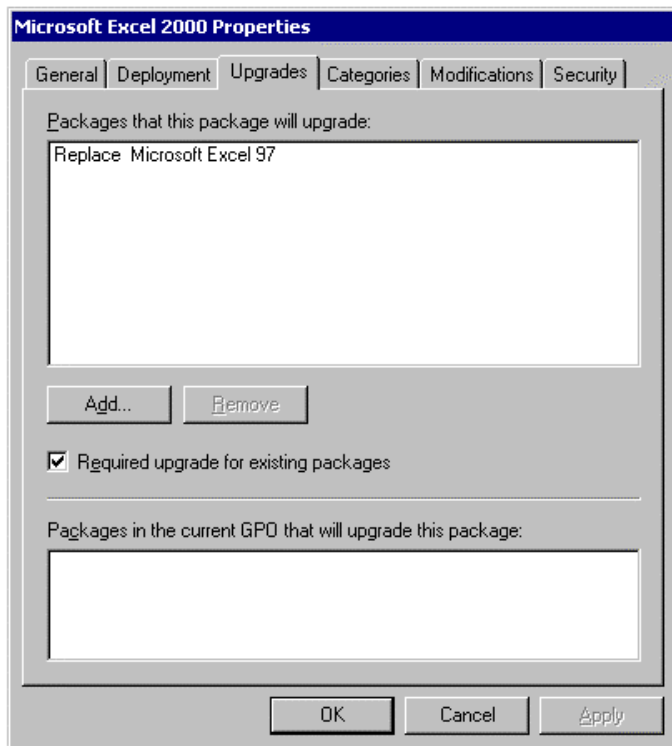
For step-by-step procedures about configuring the Auto-Install feature, see Windows 2000 Help.

#### Upgrades Tab

You begin an upgrade by using the Software Installation snap-in to assign or publish the new version and define an upgrade relationship between the new and previous deployment version. Regardless of whether the original application was assigned or published, there are two options for upgrading:

- A required upgrade happens the next time the user logs on if user-assigned, when the computer restarts if computer-assigned, or when the user goes to **Add/Remove Programs** and installs the application. After the upgrade, users who have never installed the existing version of the application can install only the new version.
- An optional upgrade does not happen immediately. Everyone who has installed the software can continue to use the existing version. Users who want the new software immediately can install it from **Add/Remove Programs**. Users who never installed the existing application can install either the existing or the new version. At some point, you might decide that it is best to make the deployed optional upgrade mandatory or required. You can go back and configure it this way. The new version of the application works like a required upgrade from that point forward.

If you want everyone who has Microsoft® Word 97 to upgrade to Microsoft® Word 2000, on the **Upgrades** tab, simply check the box **Required upgrade for existing packages**, as shown in Figure 9.9.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.9 Word 2000 Properties to Upgrade Word 97**

Looking at the application's properties dialog box in Figure 9.9, the **Upgrade** tab shows that Word 2000 is the required upgrade path for Word 97.

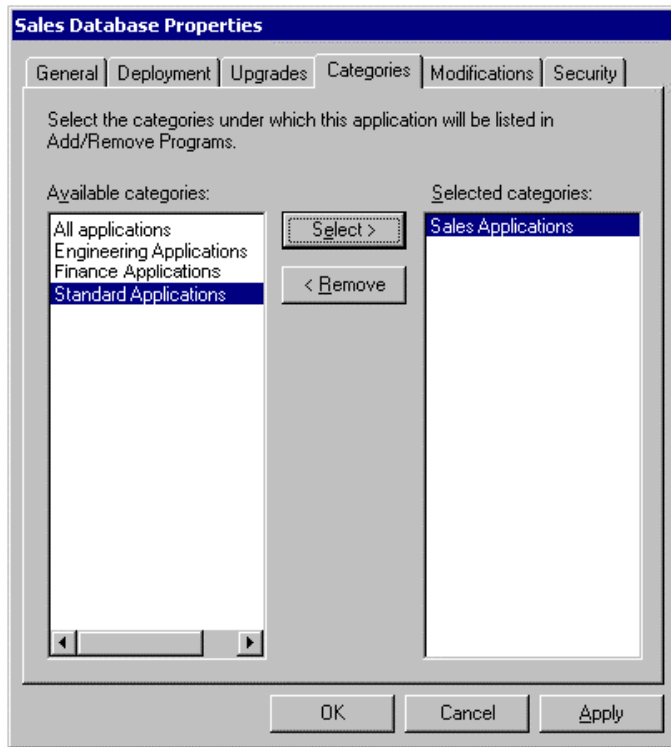
You can add Microsoft Word 2000 to the Software Installation snap-in and then you must create an upgrade relationship between Word 2000 and Word 97. Because clients can handle any number of active applications that are chained together by required upgrades, the computer will sequentially request upgrades until it achieves the current release, which is not optimal.

For more information about upgrades, patches, or services packs, see "Maintaining Software After Deployment" later in this chapter. For set-by-step procedures about configuring upgrades, see Windows 2000 Help.

#### Categories Tab

In the Software Installation snap-in, you can create and modify categories for the applications you plan to deploy. Users locate the appropriate applications in **Add/Remove Programs**. The categories that you establish are for each domain, not for each GPO. You need to define them only once for the whole domain. An administrator can choose one or more predefined categories to place an application.

Figure 9.10 shows how an administrator has selected the **Sales Applications** category from a predefined list of categories with which to associate the Sales Database application.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.10 Category Options for the Sales Database Application**

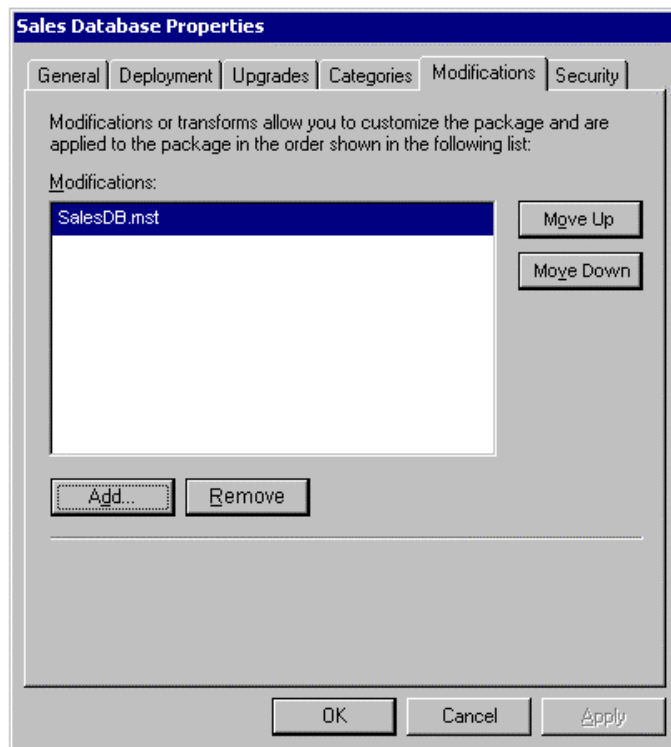
For step-by-step procedures about configuring categories, see Windows 2000 Help.

**Modifications Tab (Transforms)**

Administrators apply the .mst (transform) files to Windows Installer packages that have the .msi file name extension in an order specified by the administrator at deployment time.

You can add multiple transforms. The transforms are applied according to the order you specify in the **Modifications** list. Although settings in all transforms are merged at installation, later transforms take precedence if any settings conflict with earlier transforms. To rearrange the list, select a transform from the list and then click **Move Up** or **Move Down**.

Figure 9.11 shows the SalesDB.mst transform file in the Sales Database Properties dialog box. A sales administrator created this transform to install only certain components of the Sales Database Windows Installer package.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.11 SalesDB.mst Transform File**

**Note** It is recommended that transforms be stored at the same software distribution point and in the same shared folders as the

Windows Installer package that they customize.

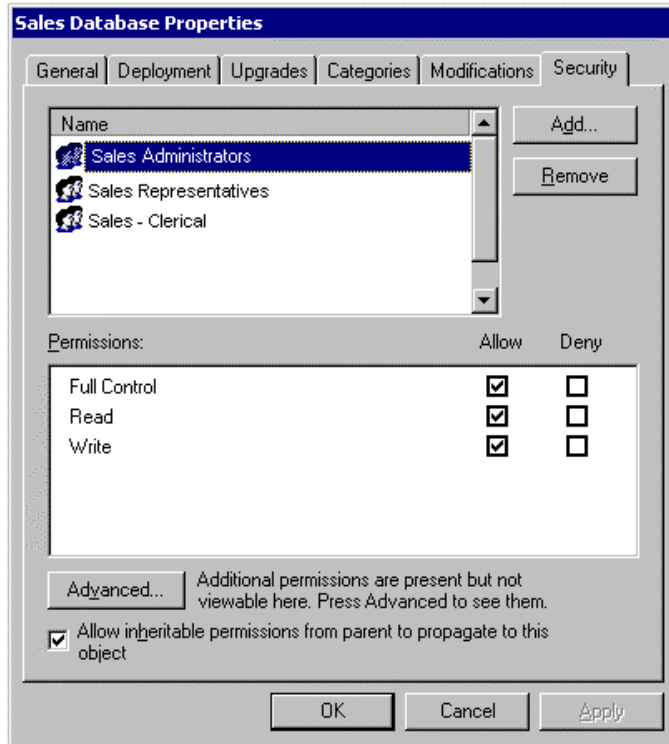
**Important** Remember that after you click **OK**, you can no longer add a transform to that deployment.

For more information about using transforms, see "Customizing Windows Installer Packages by Using Transforms" earlier in this chapter. For step-by-step procedures about using modifications, see Windows 2000 Help.

### Security Tab

The **Security** tab allows you to assign permissions to administrators and users of an application for controlled access to the software. You can also filter for each application—for example, which security groups receive an application and which do not. If you have Read access, you receive the application; otherwise, you do not. This provides much narrower control than filtering at the GPO level. If you need to change the list of security groups through which to filter a GPO, you can add or remove security groups by clicking **Add** and **Remove**. Click **Advanced** to set additional permissions.

Figure 9.12 shows Sales Administrators having Read and Write access to the Sales Database software package. If you click on the Sales Representatives or Sales - Clerical group of users, you can see that their permissions are limited to Read only.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.12** Setting Permissions for Administrators and Users

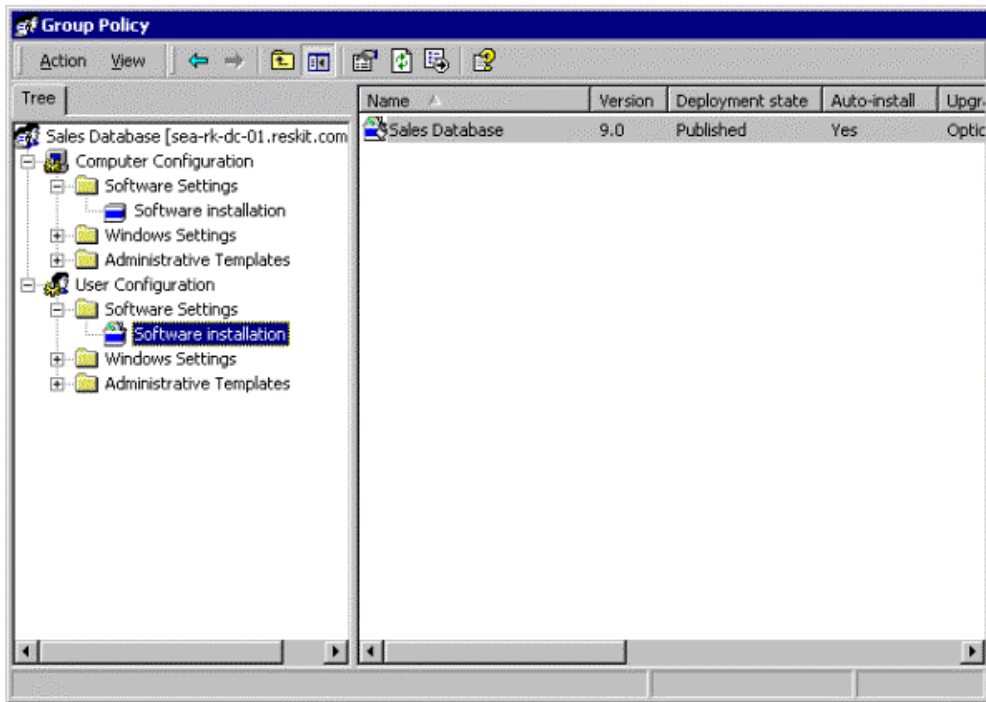
### Configuring the Software Installation Properties

In addition to the settings for each application discussed in earlier sections of this chapter, you can manage some settings for each Group Policy basis. You can manage the installation of software in the **Software Installation Properties** dialog box. The following sections provide a tour of the software installation deployment properties in the Software Installation snap-in.

#### To begin the tour of the Software Installation Properties dialog box

1. Open the GPO that you want to view or edit.
2. You will see the software root, the name appears as follows:  
*Group\_Policy\_object\_name [domain\_name]*. In the example shown in Figure 9.13, 'Sales Database [sea-rk-dc.reskit.com]' represents the GPO name and the domain name.
3. Double-click the **User Configuration** or **Computer Configuration** folder, as it applies.
4. Double-click the **Software Settings** folder.
5. Right-click the **Software Installation** folder, and then select **Properties** as shown in Figure 9.13.



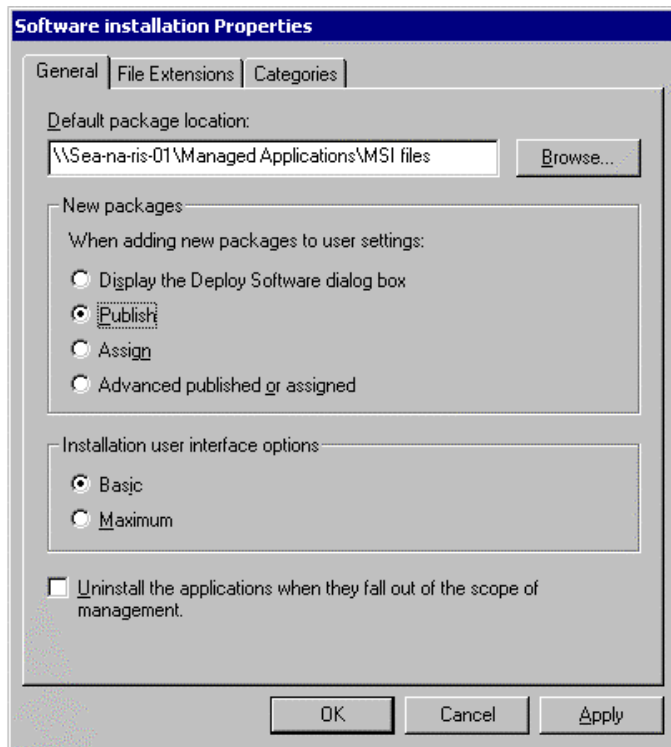


If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.13 Right-click the Software Installation Folder and Select Properties**

**General Tab**

Use the **General** tab in the **Software Installation Properties** properties sheet in the Software Installation snap-in, shown in Figure 9.14, to define the settings for each Windows Installer package you plan to deploy.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.14 Software Installation Properties**

Table 9.3 describes the options available on the **General** tab of the **Software Installation Properties** properties sheet in the Software Installation snap-in. You can use the **General** tab to define the default software distribution point where packages are stored and to determine how you want to deploy the new packages.

**Table 9.3 Options Available from the Software Installation Properties**

Areas	Options	Functions
Default package location	Identify the path to the software distribution point: \\<servername>\<sharename>	Specifies the default software distribution point.

New packages	Display the Deploy Software dialog box	Displays the <b>Deploy Software</b> dialog box for each new package so that you can choose to either assign, publish, or set advanced assign and publish properties for each package.
New packages	Publish Assign	Packages can either be published or assigned by default (to computers, assigned only). If you want to make these decisions for each package, select <b>Display the Deploy Software</b> dialog box.
New packages	Advanced published or assigned	This option allows you to get finer control on a per-package basis because it allows you to add transforms and modifications to the package.
Installation user interface options	Basic Maximum	Select one of these options to determine how much information the user sees during installation of the application: <b>Basic</b> provides only error messages and progress bars. <b>Maximum</b> shows the user any information about the installation that the creator of the package provides.
Uninstall the applications when they fall out of the scope of management	Check box	Ensures the removal of the application when the user is removed from the scope of management by this GPO.

The **Advanced published or assigned** option might be the most common selection because it allows you the most control over the package properties.

**Caution** If you assign an application to computers in a GPO that is associated with an Active Directory site and then select this option, a mobile user whose computer is managed by the GPO might lose applications when the user moves to a new site. If users that have Microsoft Office assigned to their computers travel to another site for business and Office is not assigned to computers in the new site, the application is removed when the users start their computers in the new site.

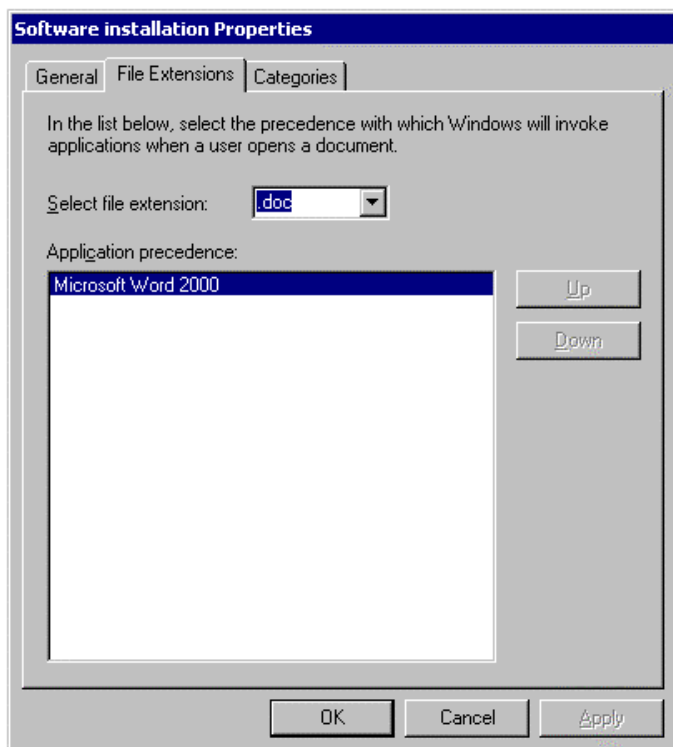
**Notes**

- Select the **Uninstall the applications when they fall out of the scope of management** check box to avoid the existence of unmanaged, orphaned applications. Also, if some applications have security sensitive information, you must make sure users can no longer use the application when they are out of scope.
- It is recommended that you begin by deploying and testing one application at a time in a pilot environment.

For step-by-step procedures about how to define the settings for a Windows Installer package, see Windows 2000 Help.

**File Extensions Tab**

In the GPO, you can select a file name extension and configure a priority for installing on-demand applications. An application is installed on demand when a user selects a file with a file name extension for a published or assigned application that is not already installed on the local computer. Configure the file name extension priority so that users who are managed by this GPO always install the correct published application when the file name extension is selected. The first application that is listed in the Software Installation snap-in on the **File Extension** tab is the application installed in association with the extension. Figure 9.15 displays the file name extension properties for Microsoft Word 2000.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.15 File Name Extensions**

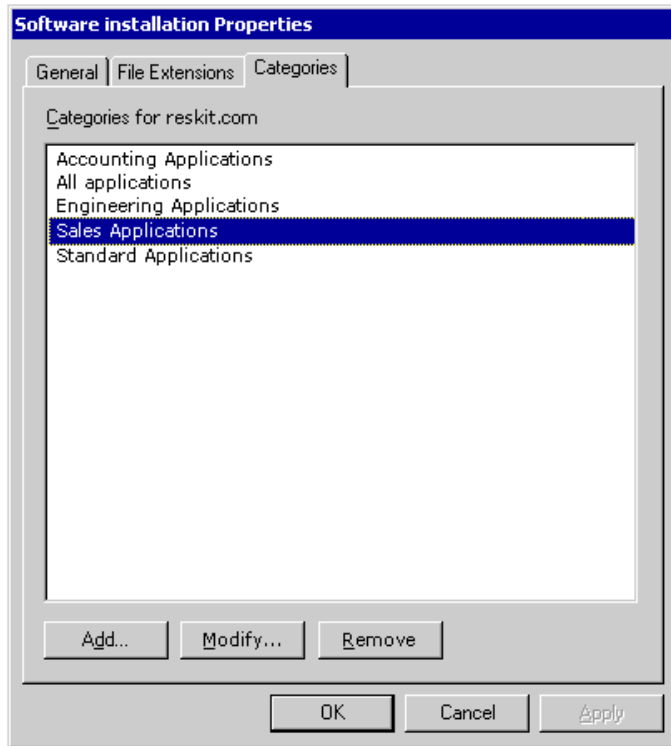
**Note** File name extension associations are managed on a per-GPO basis. Changing the priority order in a GPO affects only those users who have that GPO applied to them.

For step-by-step procedures about how to configure the file name extension priorities, see Windows 2000 Help.

**Categories Tab**

From the **Categories** tab of the **Software Installation Properties** properties sheet in the Software Installation snap-in, you can add, modify, or remove categories for the applications you plan to make available in **Add/Remove Programs** in Control Panel.

Figure 9.16 shows that the **Sales Applications** category is selected.



If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.16 Add, Modify, or Remove Categories For Applications**

For step-by-step procedures about how to add, modify, or remove categories, see Windows 2000 Help.

**Bringing Unmanaged Applications into a Managed State**

At some point it will become necessary for administrators to manage the transition of users and computers from an unmanaged environment to a managed environment. For the purposes of this discussion, an unmanaged environment is any situation where an organization is not using a Group Policy-based software management solution. Therefore, for the purposes of this discussion, using Systems Management Server represents an unmanaged situation. A managed environment is one where an organization is using Group Policy and the Software Installation and Maintenance component of IntelliMirror to manage their software.

Consider the example of an organization that wants to deploy Office 2000 on Windows NT 4.0. They know that in the future they will be deploying Windows 2000 and they will want to manage Office 2000 using Software Installation and Maintenance. With that in mind, they anticipate having to perform a management transition where they will move computers with the unmanaged Office 2000 into the managed environment. They want to ensure that they do not have to remove and then reinstall Office 2000.

Similar to when you are staging and standardizing a new computer, you want to install Office 2000 in a manner that allows it to be managed in the future. The key is to use a Windows Installer package, transform for Office 2000, and software distribution point that is exactly the same as the Windows Installer package and transform combination you will use when it is being managed.

The administrator can create the software distribution points now for use in the future, when he or she successfully transitions the software. The administrator needs to accomplish the following to ensure the eventual transition is successful:

1. Create the software distribution points and install the software with the Windows Installer packages and transforms.
2. Install the software using the Windows Installer with the following parameters:

```
\>msiexec /I \\servername\share\<software.msi> TRANSFORMS = \\servername\share\<software.mst> /qb
```

In the example above, software.msi is the Windows Installer package to be installed, and software.mst is the transform to be applied at deployment time.

3. When the managed environment is in place, including the Active Directory and Group Policy infrastructure, the administrator can assign the software in the appropriate GPO, using the same software distribution point, Windows Installer package, and transform.
4. Finally, the administrator moves the computer object into an Active Directory container with the associated GPO, or links the GPO to the Active Directory container with the computer object.

**Important** It is most important that you use the same Windows Installer package, the same transform, and the same software distribution point when deploying the application (in this example, Office 2000) in the unmanaged environment. In this case, the software will not be uninstalled and then installed again, but rather, the Application Management extension will realize that the software is the same, and it will only do what is necessary to continue to manage the software in the future.

**Note** Use the **Uninstall this application when it falls out of the scope of management** option to avoid the existence of unmanaged, orphaned applications. Also, if some applications have security sensitive information, make sure users can no longer use the application when they are out of scope.

## Deploying Software Using Remote OS Installation and Software Installation and Maintenance

When an organization first acquires new computers, IT administrators spend a lot of time preparing them for the end users. Many highly managed organizations format the hard drives of new computers, configuring it to their standard configuration.

Using Remote OS Installation services you can create a customized image of a Windows 2000 Professional desktop on a source computer. Then you can save that desktop image to the Remote OS Installation server. That image can include the operating system alone or a preconfigured desktop image, including the operating system and a standard locally installed desktop application. You can use that preconfigured image to set up multiple desktops, saving you valuable time. Create many desktop images to meet the needs of all types of users in your organization.

You can manage the applications on these computers using GPOs. However, if you want to take advantage of Group Policy-based management, some matters must be considered.

For example, consider an organization that wants to acquire new computers and install and customize software from their standard operating environment, Windows 2000 and the Office 2000 suite of applications. They want to manage these computers by using GPOs. To accomplish this, the IT administrator must:

1. Set up and configure a Remote OS Installation server.
2. Use the Group Policy snap-in to create the appropriate GPO to manage the computers associated with the RIPrep image you create. Because this is an effort to standardize all desktops in the organization, this GPO applies to all computers in the organization.
3. Place and set up a software distribution point file server for Office 2000.

**Note** Office 2000 comes with a native Windows Installer package (.msi) so you do not need to obtain a package for this application. However, if you are installing an application that does not come with a native Windows Installer package, you must obtain one. For more information about obtaining Windows Installer packages, see "Packaging Software" in this chapter.

4. Using the Custom Installation Wizard (inherent in Office 2000), you can customize Office 2000 based on users' needs. This produces a transform. For more information about transforms, see "Packaging Software" and "Installing Software" earlier in this chapter.
5. Use the Software Installation snap-in to assign Office 2000 to the computers in the appropriate GPO.

**Note** In this case, assign Office 2000 to the computers rather than the users. If you assign the software to the user, the computer-assigned version can be uninstalled, and the user version advertised and then installed on first use. While the user does acquire the software, this might take considerable time.

6. Install Windows 2000 on a new source computer and configure the operating system the way that they want it. The source computer does not have to have exactly the same hardware, but it must use the same Hardware Abstraction Layer (HAL) as the computers that will install from the image.
7. Add the computer to the Active Directory container where it will live when it is deployed. This container holds the GPO with Office 2000 assigned to the computer associated with it.
8. Restart the source computer, and then Software Installation and Maintenance installs Office 2000. That is, the software assigned to the computer installs when the computer starts.
9. Take the source computer with Windows 2000 and Office 2000 installed on it, use the RIPrep tool of Remote OS Installation to build a desktop image, and place this image on the Remote OS Installation server.

After this image is available, a user who receives a new computer (PXE-enabled) that supports Remote OS Installation server only needs to connect the peripherals (keyboard, mouse, monitor), connect to the network (plug a cable between the network card and the hub), and turn on the computer.

The computer finds the Remote OS Installation server and downloads the Windows 2000 operating system and the software. When the computer restarts after installing Windows 2000 remotely, Software Installation and Maintenance determines that Office 2000 is already on the computer, and then updates only the advertisement information. This advertisement update takes only a few seconds.

**Note** When a user logs on to the computer and selects the first Office 2000 application, they see Windows Installer start. Even though Office 2000 is already installed, it is properly separating installation from user configuration. The Windows Installer starts each time a new user starts Office 2000 to complete a small amount of user configuration. This happens with Office 2000 regardless of whether Office 2000 is assigned to the user, assigned to the computer, published for the user, or installed by using Remote OS Installation.

Remote OS Installation and Software Installation and Maintenance together enable administrators to rapidly and efficiently deploy both the operating system and other applications, and still bring the software into a state where it can be managed by Software Installation and Maintenance.

For more information about Remote OS Installation, see "Configuring Remote Operating System Installation" in this book.

## Deploying Software Using SMS

Microsoft® Systems Management Server (SMS) version 2.0 supports a distribution model that you can use with Software Installation and Maintenance. For enterprise-sized organizations, you can use SMS to analyze your network infrastructure and distribute software packages to the numerous software distribution point file servers. You can then use Software Installation and Maintenance to distribute the software to all Windows 2000-based clients. SMS provides full distribution support including compression, bandwidth throttling, priorities, checkpoint restart, scheduling, load balancing, and replication.

For more information about using SMS, see "Using Systems Management Server to Supplement IntelliMirror" in this book.

## Using Terminal Services for Remote Administration

You can set up a Terminal Services server to run either in a remote administration mode or an application server mode. Remember that:

- Remote administration mode allows a user to use the Terminal Services server to perform only administration functions.
- Application server mode provides multiple concurrent user connections to any number of applications.

Using Terminal Services, you can deploy and manage applications from a central location. After an application is deployed, many clients can connect to the Terminal Services server. Applications can be installed on clients from Terminal Services servers configured as remote administration servers using Windows Installer. The underlying Terminal Services server code and Windows Installer code work together to deliver an installation, in contrast to how applications were installed on a Terminal Services server in the past. Then the computer had to be put in an installation mode first; and after application installation it was put back into an execution mode.

Table 9.4 shows when assigned and published software is supported on remote administration and application Terminal Services servers.

**Table 9.4 Support for Managed Software on Terminal Services**

Software Installation Method	Remote Administration	Application Server

Published	Supported for both Windows Installer packages and existing setup programs defined in .zap files for publishing.	Not applied, software is not installed.
User assigned	Supported: Software Installation and Maintenance works in the same manner as on Windows 2000 Professional.	Not applied, software is not installed.
Computer assigned	Supported	Supported: A domain user with a roaming user profile might roam to an application server. Their application shortcuts follow them to the application server. If the server has the same application either installed (per computer) or assigned, and the user activates the shortcut, the shortcut works (either activating or installing the application). If the application is neither installed nor assigned, the shortcut does nothing.

On an application server, software installation must not jeopardize the running system, and it must be configured to allow concurrent usage of an application. For these reasons, only an administrator can permit software installation. To accomplish this, Terminal Services servers that are configured as application servers restrict application deployment. Only administrators are allowed to initiate a Windows Installer–based application installation, and the application must be fully installed before users can use the computer by means of remote sessions. It is your responsibility to decide what is required, and to ensure that all bits are locally installed and available prior to allowing remote user connections.

### Maintaining Software After Deployment

After the software is installed, the maintenance phase begins. Maintaining software in your IT infrastructure includes installing patches or upgrades and removing software altogether.

### Updating Existing Software Using Patches and Upgrades

From time to time, publishers of software provide patches or hotfixes, service packs, and software upgrades.

Typically patches address a single problem and contain the same product code or an .msp file. You need to determine whether or not your organization needs the patch and if so, whether you are going to manage it. Usually, when you deploy patches or minor software updates, you re-advertise the package to everyone who was granted access to the original application.

#### Patching Applications

From time to time, authors of software provide patches. Patches typically fix a single problem, and the amount of testing that is done on a patch will vary. Depending on the complexity of the patch, you might need to deploy the patch as an upgrade. The number of files the patch will affect is the determining factor. Through testing, administrators determine whether their organization requires the patch and whether they will deploy it as a patch or an upgrade.

After an administrator tests a patch and decides to deploy it as a patch, he or she will copy the patched files to the software distribution point, replacing the older files. The author of the software patch either supplies a new Windows Installer package (an .msi file) or a Windows Installer patch (an .msp file). If the author supplies a new Windows Installer package, the administrator simply replaces the existing package at the software distribution point with the new package. However, if the author supplies a Windows Installer patch, the administrator needs to apply the patch (.msp) to the existing package.

After the files at the software distribution point are updated with the new, patched versions, the administrator then opens the Software Installation snap-in within the GPO where he or she is managing the existing software. Then he or she clicks the software that they just patched at the software distribution point, and then selects **Redeploy Application** from the context menu.

For step-by-step instructions about patching applications, see Windows 2000 Help.

#### Upgrading Applications

Upgrades typically involve major changes to the software and have new version numbers. A substantial number of files might change for an upgrade. You can use the Software Installation snap-in to establish the procedure for upgrading from an existing application to the current release.

In the case of a natively authored package, the author of the software supplies a Windows Installer package. That package defines which existing versions of the software the new package can upgrade. It also contains instructions about how to perform the upgrade — for example, which existing files can be left in place, which existing files must be deleted, and which new files need to be installed. A native Windows Installer package lists the other applications the package knows how to upgrade. You can use the list the Software Installation snap-in provides to help you to manage upgrades.

You can use the Software Installation snap-in to manually create upgrade relationships with other packages in your organization that the application replaces. This includes making a formal upgrade relationship between two similar products from completely different vendors. To summarize, you can replace one vendor's application with another or you can upgrade a repackaged application.

Repackaged applications that are deployed by using the Software Installation snap-in can impact upgrades in the following ways:

- A repackaged application's Windows Installer package does not have declared upgrade relationships. You must manually create upgrade relationships, which the Software Installation snap-in supports.
- During an upgrade, it might not be possible to completely remove a repackaged application. The removal of a repackaged application might leave components on the desktop, even if the component is neither shared nor needed. You can create a script to clean up the unnecessary, leftover files.

**Note** When a repackaged application is upgraded, the existing application is removed, and the new version is installed, this might result in the loss of user preferences and individual settings. However, if you replace one native Windows Installer package with another (or if you do an in-place upgrade), this problem does not occur because a native package only replaces files that need to be replaced, and does not automatically remove all registry entries and other files where user preferences might be stored.

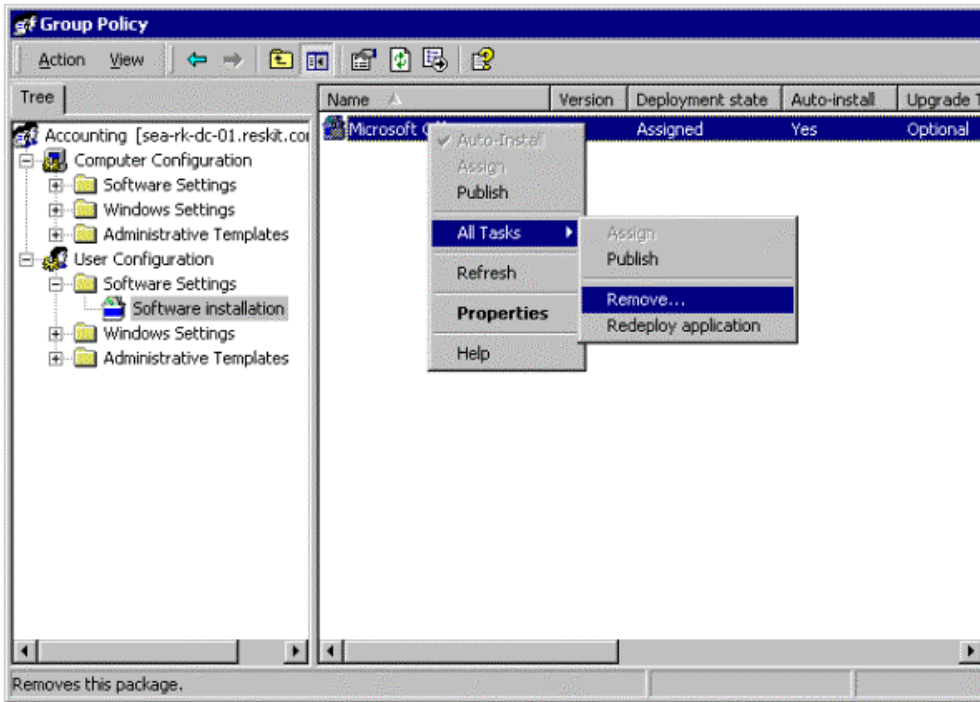
It is recommended that you pilot or test an upgrade before putting it into production. In the pilot phase, you can give the users the new version of an application and retain the original. You can set up two separate GPOs for each software package.

For step-by-step instructions on upgrading applications, see Windows 2000 Help. For information about the Software Installation and Maintenance upgrade configuration options, see "Installing Software" earlier in this chapter.

### Removing Software

At some point, you need to remove the software. Figure 9.17 shows you how to access the **Remove Software** options menu dialog

box in the Software Installation snap-in.



If your browser does not support inline frames, [click here](#) to view on a separate page.

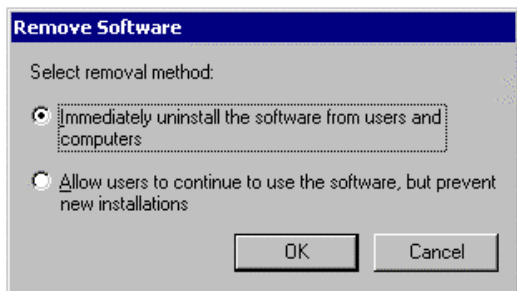
**Figure 9.17 Accessing the Remove Software Option Menu**

The following choices, set within the Software Installation snap-in, might affect the removal of software:

- Optional Removal.** You decide the existing software is not supported and that Help desk support is provided only for the latest version of the software.

You can remove the software from management without forcing the (physical) removal of the software from the computers of users who are still using the software. In this situation, no one can install the older version of the software from the **Start** menu, by using **Add/Remove Programs**, or by document invocation. However, users who already have the software installed can continue to use the software until they remove it themselves.
- Forced Removal.** You decide that it is best that the software no longer be used. That is, you do not want users to install or run the software. With forced removal, the software is automatically removed from a computer, either the next time the computer is turned on or restarted (when the software is assigned to the computer) or the next time the user logs on (when the software is assigned or published to the user).

Figure 9.18 shows the software removal options you can select to meet the needs of your users.



**Figure 9.18 Remove Software**

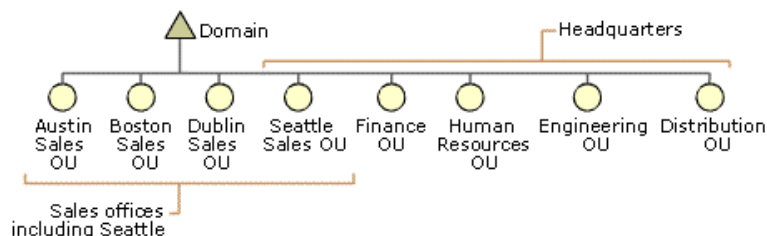
**Note** When you originally deploy the software, if you want the application to be removed when a GPO is no longer applicable, select the **Uninstall this application when it falls out of the scope of management** option. For more information about this option, see "Targeting Software" earlier in this chapter.

**Software Installation and Maintenance Deployment Situations**

The following example situations demonstrate how an enterprise-sized manufacturing firm uses the Software Installation and Maintenance component of IntelliMirror to install and manage software. These situations assume that IT administrators have performed all testing prior to installing any software.

Figure 9.19 shows a grouping of OUs that pertain to the example situations that follow. This flat OU design is only an extract from a larger domain infrastructure.





If your browser does not support inline frames, [click here](#) to view on a separate page.

**Figure 9.19 Extract from the Manufacturing Firm's Complete Domain Infrastructure**

#### Situation 1: Deploying a New Application

The sales personnel of the manufacturing firm require two new sales-based applications: a sales database and an order entry application. The sales personnel include outside sales representatives, inside sales representatives, sales management, and clerical sales staff. These users are dispersed across four organizational units in the domain infrastructure. Each organizational unit is named for the sales office locations: Austin Sales, Boston Sales, Seattle Sales, and Dublin (Ireland) Sales. Each organizational unit consists of up to 150 users.

The IT administrator, who manages the sales OUs, plans to publish the two new sales applications for all sales personnel. The following is the process the IT administrator follows to deploy the new sales applications:

1. **Obtains the Windows Installer Packages.** The new sales software comes complete from the manufacturer with Windows Installer packages (SalesDB.msi and OrderEntry.msi). Each application is less than 50 MB.  
**Note** If the author of these applications did not provide the .msi files, the administrator needs to repackage the software into two separate Windows Installer packages.
2. **Places the software distribution point file servers as close as possible to the users.** In this case, the IT administrator uses the existing software distribution points in each of the U.S. sales offices because they are only 20 percent used. With regard to the Dublin, Ireland sales office, the IT administrator sets up a new software distribution point file server because none currently exists.
3. **Creates the file shares on the software distribution points and copies the packages to the shares.**
4. **Creates and applies a GPO.** The IT administrator creates a new GPO named Sales Personnel, and then specifies that these two applications be published to the members of the sales office OUs.
5. **Sets up DFS to manage the network traffic.** The IT administrator employs DFS to manage the network traffic during software installation time. This helps balance the software installation traffic and protects against bottlenecks.
6. **Breaks up the software deployment into phases.** To minimize network traffic, the IT administrator decides that she wants to break up the software deployment into four phases. The deployment is performed Monday through Thursday for up to 150 users per day.

Instead of deploying the applications to all users in each OU at the same time, she will deploy the packages to a fraction of the users in each organizational unit. For example, because each sales office OU contains about 100 to 150 users, the administrator breaks two OUs into four virtual groups of 50 to 75 users. On Monday the administrator sends half of the users in Austin and half of the users in Boston an e-mail message telling them that the new applications, "Sales Database" and "Order Entry," are now available for installation from **Add/Remove Programs** under the **Sales Applications** category. The following day, the administrator covers the other half of users in the Austin and Boston sales offices. The administrator follows the same process for all sales offices until all sales personnel are informed of the new software.

7. **Informs the users of the availability of the new sales applications.** When the administrator informs the users of the availability of the new software, she also provides them with instructions on how to access **Add/Remove Programs** to install the applications.

The users download the applications from the **Sales Applications** category in **Add/Remove Programs**.

8. **Supports both the newly installed software and the soon-to-be retired software.** While the deployment is progressing, IT administration supports both the previously installed sales applications and the newly deployed sales applications.
9. **Allows a grace period for people to upgrade to the new version.** The IT administrator informs the users that the old sales database and order entry applications will no longer be available to them at least 30 days after the new software is completely deployed. This encourages all sales personnel to install and use the new software so that when the previous sales applications are removed, the users no longer depend on them. It is good business practice to give the users plenty of notice when you plan to remove software on which users had heavily relied.

#### Situation 2: Performing an Upgrade to Remove and Replace Software

The manufacturing firm signs a global licensing agreement with a new anti-virus software provider. Currently, all computers in the organization have X anti-virus software installed. By using Software Installation and Maintenance, IT administrators safely replace X anti-virus software with Y anti-virus software.

Anti-virus software is critical to any network operating environment and therefore, IT administrators assign Y anti-virus software to all computers in the organization. The following is the process the IT administrator follows to replace X anti-virus software with Y anti-virus software:

1. **Opens the Software Installation snap-in from within the GPO.** The GPO includes the Software Installation snap-in that contains application deployment properties of X anti-virus software.
2. **Makes an upgrade relationship between X anti-virus software and Y anti-virus software.** On the **Upgrade** tab of the **Y Anti-Virus Properties** dialog box, the administrator selects the **Required upgrade for existing packages** check box. This ensures that all users of X anti-virus software will receive the upgrade to Y anti-virus software.
3. **Removes X anti-virus software and replaces it with Y anti-virus software.** On the **Deployment** tab of the **Advanced Deployment Options** properties sheet, which also resides in the **Y Anti-Virus Properties** dialog box, the administrator selects the **Remove previous installations of this product** check box. This safely removes the X anti-virus software and replaces it with Y anti-virus software.
4. **Prompts the users to restart their computers.** Prompt the users by means of e-mail to restart their computers so that the computers will recognize the new software.

After the users restart their computers, the computers recognize the new Y anti-virus software.

#### Situation 3: Performing an In-Place Upgrade

For years, everyone at the manufacturing firm has been using a mission-critical, custom application specific to their field of business. Recently, developers were recruited to redevelop the application to include many improvements, one of which includes a native

Windows Installer package. IT administrators now want to deploy the new .msi package. They have chosen to perform an "in-place upgrade." The following is the process the IT administrators follow when performing the in-place upgrade:

1. **Determines that the properties for the previously installed custom manufacturing application are set correctly for the new deployment.** Open the Software Installation snap-in from within the GPO. The GPO includes the Software Installation snap-in that contains application deployment properties for the previously installed custom manufacturing application. If necessary, make changes to the application properties name on the **General** tab in the Software Installation snap-in properties sheet.
 

**Note** This software is mission-critical for all users in the manufacturing firm and therefore, it has been and will continue to be, assigned to everyone in the organization.
2. **Copies the new .msi package upgrade to the software distribution points.** The software distribution points detect the new package to be a completely different software package from the previously deployed version of the custom application. The IT administrator copies the new .msi package to all but one software distribution point.
 

**Note** When creating a software distribution point for a new application, you must specify a different DFS path even if the underlying replicas are on similar shares. That way, you can retire software distribution points for the old software by taking down replicas of the DFS tree while you keep all the new software distribution points of the new application up due to the fact that it is in a different DFS path.
3. **Allows a grace period for people to upgrade to the new version.** It is good business practice to give the users plenty of notice when you plan to remove software on which users had heavily relied.
4. **Copies the new .msi package upgrade to the final software distribution point.** After you are certain that all users have upgraded, you can upgrade the final software distribution point and retire the previous version.
5. **Prompts the users to restart their computers.** Prompt the users by means of e-mail to restart (if computer assigned) or log off (if user assigned or published) their computers so that the computers will recognize the new software. After the users restart their computers, the Group Policy setting is applied and the users have access to the new, improved custom manufacturing application.

#### Situation 4: Patching an Existing Application

The manufacturing firm's corporate headquarters' address has changed and IT administrators want to deploy a Word 2000 fax and letterhead template patch to replace the previous fax and letterhead template files. With the exception of the sales offices, the OU structure is designed around the organization's departmental structure. With this in mind, the IT administrator needs to selectively deploy the fax and letterhead templates to the departmental OUs and the Seattle-based sales office. The sales offices in Austin, Boston, and Dublin need not receive the patch. The following is the process the IT administrator follows to patch some previously deployed software:

1. **Copies the patched (.msp) file to the applicable software distribution points.** Doing this replaces only the previously deployed fax and letterhead files of Word 2000 on each file server.
 

**Note** If the patched file is an .msi file, the administrator replaces the existing package at the software distribution point with the new Windows Installer package.
2. **Opens the Software Installation snap-in from within the GPO.** The GPO includes the Software Installation snap-in that contains application deployment properties of Word 2000.
3. **Redeploys the fax and letterhead files.** From within the GPO, right-click **Word 2000**, select **All Tasks**, and then select **Redeploy Application** from the context menu. This works only in the cases of computer assigned applications.
4. **Prompts the users to restart their computers.** Prompt the users by means of e-mail to restart their computers so that the computers will recognize the patched software. After the users restart their computers, the GPO applies the patched file and the users have access to the new fax and letterhead templates.

#### Situation 5: Removing Software

The manufacturing firm suddenly terminates a site license agreement with a certain software distributor for a marketing application that was previously published to the users in the Marketing department. IT administrators of the firm must ensure that all copies of the software are removed. The following is the process the IT administrator follows when removing (in this case, unlicensed) software:

1. **Opens the Software Installation snap-in from within the GPO.** The GPO includes the Software Installation snap-in that contains application deployment properties of the marketing application.
2. **Removes the marketing application.** From within the GPO, right-click the marketing application, select **All Tasks**, and then select **Remove** from the context menu.
3. In the **Remove** dialog box, select the **Immediately Uninstall** check box.
4. **Prompts the users to restart their computers.** Prompt the users by means of e-mail to restart their computers so that the unlicensed software is no longer available for use. After the users restart their computers, the GPO applies and the users no longer have access to unlicensed software.

---

[Send feedback to Microsoft](#)

[© 2004 Microsoft Corporation. All rights reserved.](#)