realtimepublishers.com™

*The How-To Guide™ To*

# Windows Server 2003 Terminal Services

triCerat®

*Greyson Mitchem*

## Copyright Statement

# Chapter 2: Application Installation and Configuration

Starting with the release of Windows 2000 (Win2K), Microsoft has included terminal server compatibility in its Certified for Windows logo program, so most current applications can be installed and run in the multi-user environment of a terminal server without modification. However, you should still be familiar with the process of installing applications on a terminal server as well as the application-compatibility subsystems in case you encounter a legacy program that you must integrate into your terminal server environment.

The Windows logo certification specification instructs programmers to take advantage of several Windows component services that would make the application natively compatible with WS2K3 and Terminal Services. The following list quotes and describes the elements of the specification that are of particular interest to the Terminal Services administrator:

- *Do not read from or write to Win.ini, System.ini, Autoexec.bat, or Config.sys on any Windows operating system based on NT technology*—Programs that don't obey this rule might store per-user settings in these per-machine configuration files.
- *Install using a Windows Installer-based package that passes validation testing* and *Ensure that your application supports advertising*—The Windows Installer service uses a process called *advertising* to ensure that per-user registry keys and files are installed for each user of a computer and not just the user who installed it.
- *Default to My Documents for storage of user-created data*—Compliance with this item ensures that per-user files (documents, macros, templates, and so on) are stored in a per-user location, not in the program's directory.

In the real world, however, systems administrators have to deal with many applications—both current and legacy—that don't adhere to these guidelines or were written before the specification was established. To assist in integrating these types of applications, Terminal Services utilizes several application compatibility subsytems.

## Application Compatibility Subsystems

There are three main application compatibility subsystems: registry mapping, INI file mapping, and root drive. The subsystems have two modes—install and execute. Install mode is used during the installation of user applications, and execute mode is the normal operating mode for the terminal server.



It is often tempting to install Terminal Services purely to overcome the two session limitation imposed by Remote Desktop. You need to be aware that in addition to eliminating the limit on the number of sessions available, Terminal Services enables these application compatibility subsystems, which may effect the way that server-based applications behave.

### Registry Mapping

During installation, many applications add registry information to the Current User registry hive (HKCU). If the application does not take advantage of the Windows Installer Service, or have its own way of populating HKCU keys upon launch, only the person installing the application will have the correct values. In a workstation environment, this consideration is not usually an issue as the application is installed under the context of the user that will run it. On a terminal server, however, such is not the case, as many users will run the application on the same server.

Registry mapping takes care of this problem. While in install mode, the terminal server monitors any write actions to HKCU during the install process and replicates the keys to a special subsection of HKLM. Then, while in execute mode, the terminal server monitors read requests to HKCU by the application. If the application attempts to read from a key that does not exist, the terminal server will look in HKLM for the values and copy them up to HKCU before the application is aware that they are missing. The repository for registry mapping data is HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software.

> ☞ If you want to modify the default values for an application that uses registry mapping, simply add or edit values in the just-mentioned HKLM key.

### INI File Mapping

Some applications continue to store settings (either per-machine or per-user) in INI files instead of in the Windows registry. This action poses a challenge on a terminal server, as no single user can have exclusive access to the INI file, nor can per-user settings be stored there, as changes made by one user will affect everyone on the server.

INI file mapping allows each user to have his or her copy of the INI file without having to recode the application to look in a user-specific location for the file. While in install mode, the terminal server monitors modifications to the WIN.INI file as well as the creation of new INI files in the Windows or Program Files directory. Then, in execute mode, the terminal server will create copies of the INI files in either the user's profile or the user's home directory. When the application attempts to read from or write to the original file, the subsystem redirects the action to the user's copy. Changes to INI files are handled by comparing the date stamp on the user's copy with that of the original file, and if the original is found to be newer, the two files are merged to create a new file for the user.

### Root Drive

Older versions of Windows were not able to map drives to a subdirectory of a share. A user's home directory, for example, would be represented by H:\%username% and not simply H:\ as the server could only map to the home share itself. This shortcoming posed a challenge, as many applications do not allow for system variables when referencing files. The root drive concept was developed to create a uniform path that referenced a per-user location. This way, you could simply reference the root drive letter (R:\ perhaps) and have the destination be either the user's specific home directory or user profile.

During logon, a script called usrlogon.cmd is run on all terminal servers. If root drive has been enabled on the server, the script performs a SUBST command that aliases the user's home directory to the defined root drive letter.

> ☞ The SUBST command works much like the NET USE command, but instead of aliasing a network path to a drive letter, SUBST aliases a local path. The SUBST command can also be used to alias a subfolder of an existing *mapped network drive* to another drive letter.

Starting with Win2K, Windows is able to map drives to subfolders of network shares, so a user's home directory can be represented by H:\ even if it is a subfolder of the HOME share. This feature has made the use of a root drive almost obsolete. Many existing application compatibility scripts, however, are still written to reference %rootdrive%. Thus, you may still need to define and use a root drive. You can, however, modify the userlogon.cmd file on your server to take advantage of the existing drive letter instead of using the SUBST command to alias another letter to the same location. Listing 2.1 provides an example of how you can do so (changes are in red).

```
Cd /d %SystemRoot%\"Application Compatibility Scripts"
Call RootDrv.Cmd

If "A%RootDrive%A" == "AA" goto done

REM If the user has a network Home Directory already mapped
REM on the ROOTDRIVE, we do not need to do anything.

if /I "%rootdrive%" == "%homedrive%" goto NoSubst

:DoSubst
Net Use %RootDrive% /D >NUL: 2>&1
Subst %RootDrive% "%HomeDrive%%HomePath%"
if ERRORLEVEL 1 goto SubstErr
goto AfterSubst
:SubstErr
Subst %RootDrive% /d >NUL: 2>&1
Subst %RootDrive% "%HomeDrive%%HomePath%"
:AfterSubst

:NoSubst
```

***Listing 2.1: Modified USRLOGON.CMD.***

## How to Toggle Between Install and Execute Mode

Both registry mapping and INI file mapping have different behaviors based on whether the terminal server is in install or execute mode. You should be careful to make sure that the server is in the proper mode at all times. To switch the server to install mode, open a command shell (cmd.exe) and type:

```
change user /install
```

To switch the server to execute mode, open a command shell and type:

```
change user /execute
```

To determine which mode the server is currently in, open a command shell and type:

```
change user /query
```

You can use any of these commands in shell scripts (batch files) as well.

Alternatively, to have the server automatically switch between install and execute modes, use the Add/Remove Programs Control Panel applet. To do so, click Add New Programs, then click CD or Floppy (see Figure 2.1). Doing so will start the Install Program wizard.

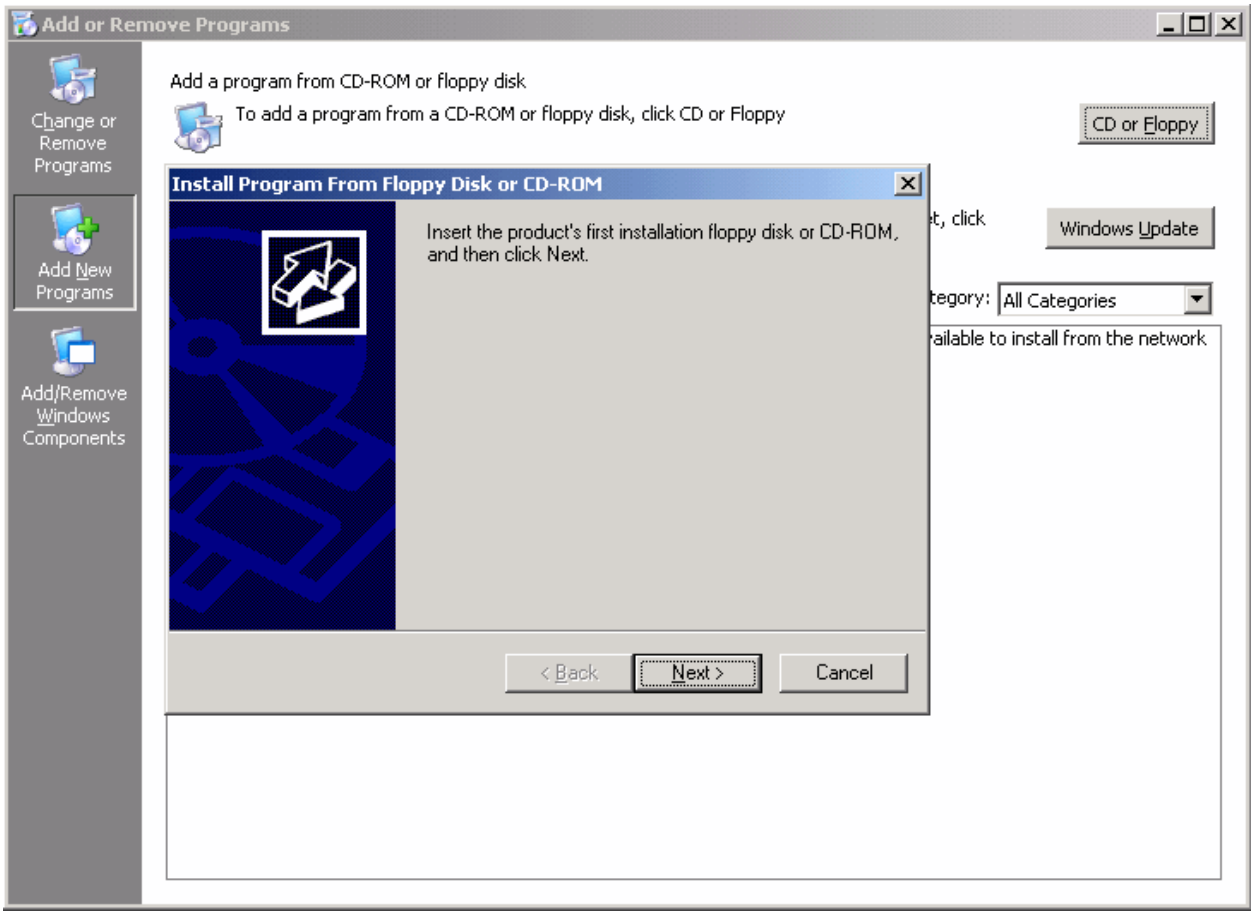


***Figure 2.1: Activating install mode via the Add/Remove Programs Control Panel applet.***

After you select the setup program, the wizard will automatically switch the server to install mode and run the installation program. When the installation is complete, click Finish in the wizard, and the server will switch back to execute mode.

> ☞ After a reboot, the server will always come up in execute mode, regardless of which mode it was in when it was shut down.

WS2K3 will detect most installation packages and automatically invoke the Install Program wizard for you. However, it is still a good idea to use the Control Panel or command line when installing new applications just to be safe.

## Windows Installer Service

The Windows Installer Service (msiexec.exe) and its ability to "advertise" settings has virtually eliminated the need for registry mapping. When installing applications that come packaged in MSI format, you usually do not need to place the server into install mode.

> Some applications are not packaged to take full advantage of MSI advertising, so you might want to continue to use install mode just to be safe.

When a user launches an "advertised" application, the Windows Installer service is invoked and any per-user settings are added to HKCU. The package can even be set up to add per-user files to the user's profile under application data or local settings. This functionality eliminates the need for many application compatibility scripts.

## How to Install MSI Packages on Terminal Servers

To install an application that has been packaged in MSI format, simply double-click the MSI file. Use the installation wizard to select the components you want to install.

> ☞ If the installation wizard asks you whether you want to install the application for yourself or for all users, be sure to select "All users" as doing so will register the components to be advertised to other users.

You can also install MSI packages via the command line by using the msiexec command. Later, this chapter provides a complete list of arguments used to install, uninstall, and repair packages.

### *Via IntelliMirror/Group Policy*

If you are managing a large terminal server farm or need the ability to expand your farm quickly, you will want to use an automated software installation technology to install MSI packages on terminal servers. In an AD environment you can use Group Policy to assign applications to your servers. When the server boots, the machine policies are processed and any applications that are assigned will be installed.

To use Group Policy to install applications, your applications must be in MSI format. Group Policy also supports a script format called ZAP, which can trigger a non-MSI installer. However, the use of this format is not recommended with terminal servers because such applications do not support advertising, so your users may not receive necessary HKCU registry keys.

> You can repackage software into the MSI format by using a third-party utility such as Wise for Windows Installer. Be sure to thoroughly test the application on Terminal Services before and after repackaging to determine whether any modifications need to be made for terminal server compatibility.

The basic steps required for Group Policy-based software installation are:

1. Create a network share to store your MSI packages.
2. Create Administrative Installations of your MSI packages and place them on the share.
3. Add the packages to a GPO that applies to the terminal server computers in AD.
4. Modify the permissions on the packages in the GPO if you want to filter which terminal servers receive each package.
5. Reboot the terminal servers.

### Create a Share

To assign or publish an application via Group Policy, you need a central location to reference for the application source files. The path to this location must be resolvable and accessible by all computers to which the policy applies. The easiest way to accomplish this task is to create a share on a file server and copy the source files to it.

> Make sure that the machine accounts of your terminal servers have read and execute rights on the share. The Authenticated Users and Everyone groups both include machine accounts.

### Create Administrative Installations

MSI packages typically come with all the files needed for the application compressed into CAB files. To optimize the installation of the software, uncompress the files in advance by creating an Administrative Installation.

To create an Administrative Installation, execute the Windows Installer Service with a "/a" switch, and specify the path and name of the MSI package you want to uncompress:

```
msiexec /a d:\proplus.msi
```

A wizard, similar to the one that Figure 2.2 shows, should appear asking you for the destination directory for the Administrative Installation. If the application requires a license key for installation, the wizard will prompt you for this as well. The key is then encrypted into the Administrative Installation so that installs performed from that source will not be prompted for the key.

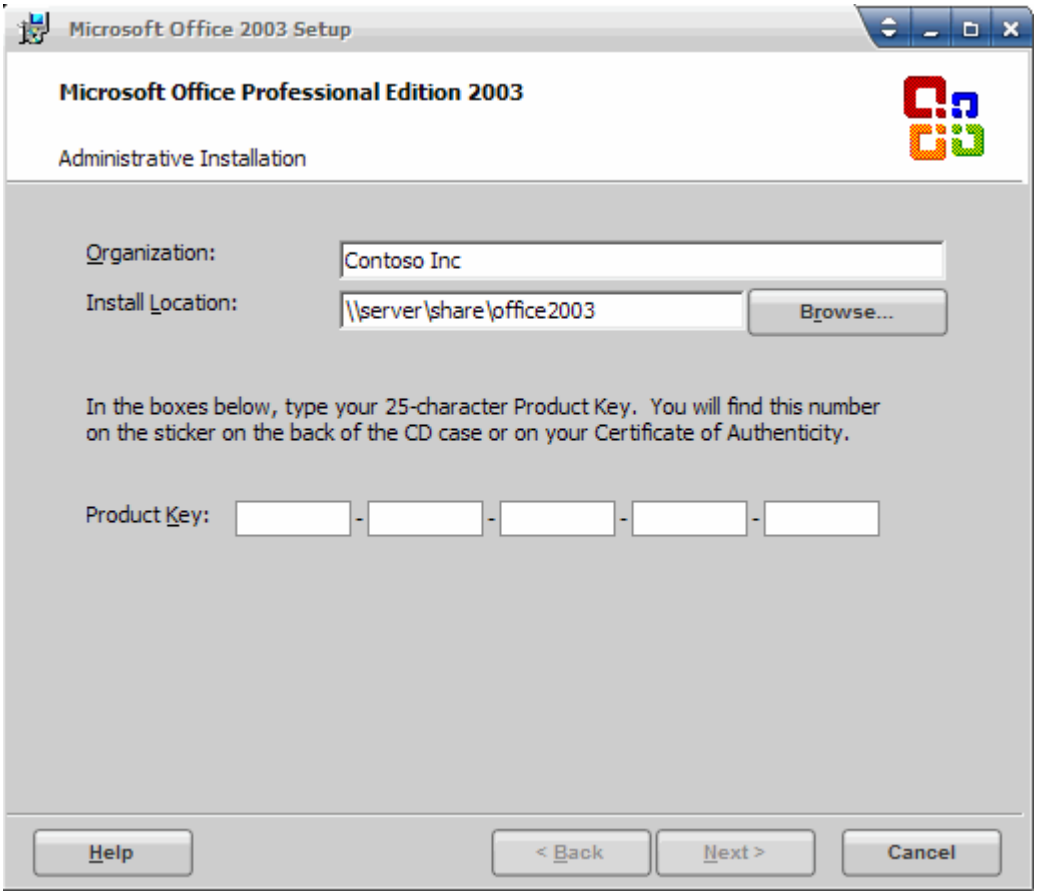


***Figure 2.2: An Administrative Installation wizard.***

Once the Administrative Installation is complete, copy the new source files to your share.

## Add the Packages to a GPO

To add a package to a GPO, edit the policy, and expand Computer Configuration, Software Settings. Right-click *Software installation*, and select New, Package (see Figure 2.3).

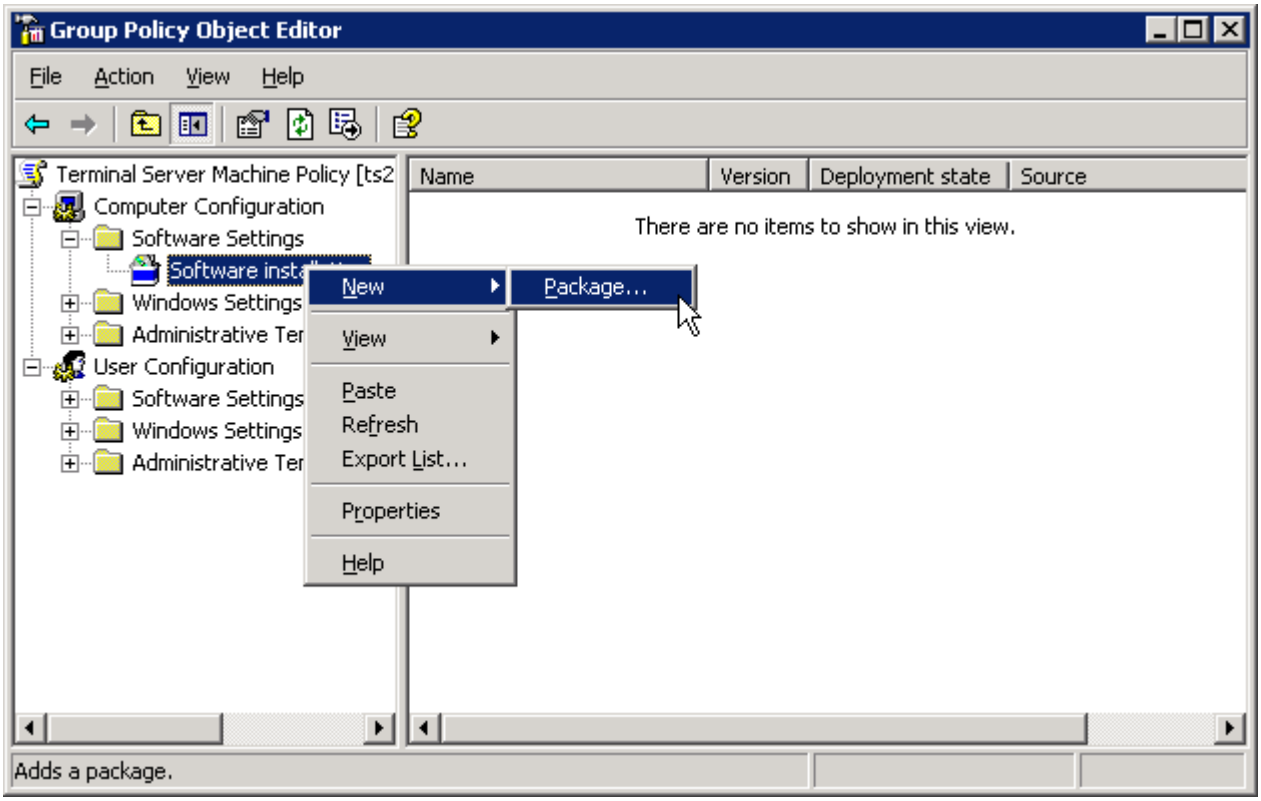


***Figure 2.3: Adding a software package to a GPO.***

You will be prompted for the MSI package that you want to add. Navigate to your share and select the MSI file that you want to install. You will then be asked whether you want to assign the package or open the Advanced Settings interface. In most cases, you can select Assign and accept all the default options. If, however, you need to specify a transform to be applied during the installation, select Advanced.

> A transform (MST file) specifies options or makes changes to the default settings of a Windows Installer package (MSI file). You can create transforms by using the Microsoft Office Custom Installation Wizard or with a third-party utility.

## Filtering Applications

You can use a single GPO that applies to all your terminal servers and still filter which applications are installed on each server by using security filters on the packages in the GPO. Figure 2.4 shows a GPO that contains three software packages—Adobe Acrobat Reader 6.0, Office XP, and the Microsoft Group Policy Management Console.

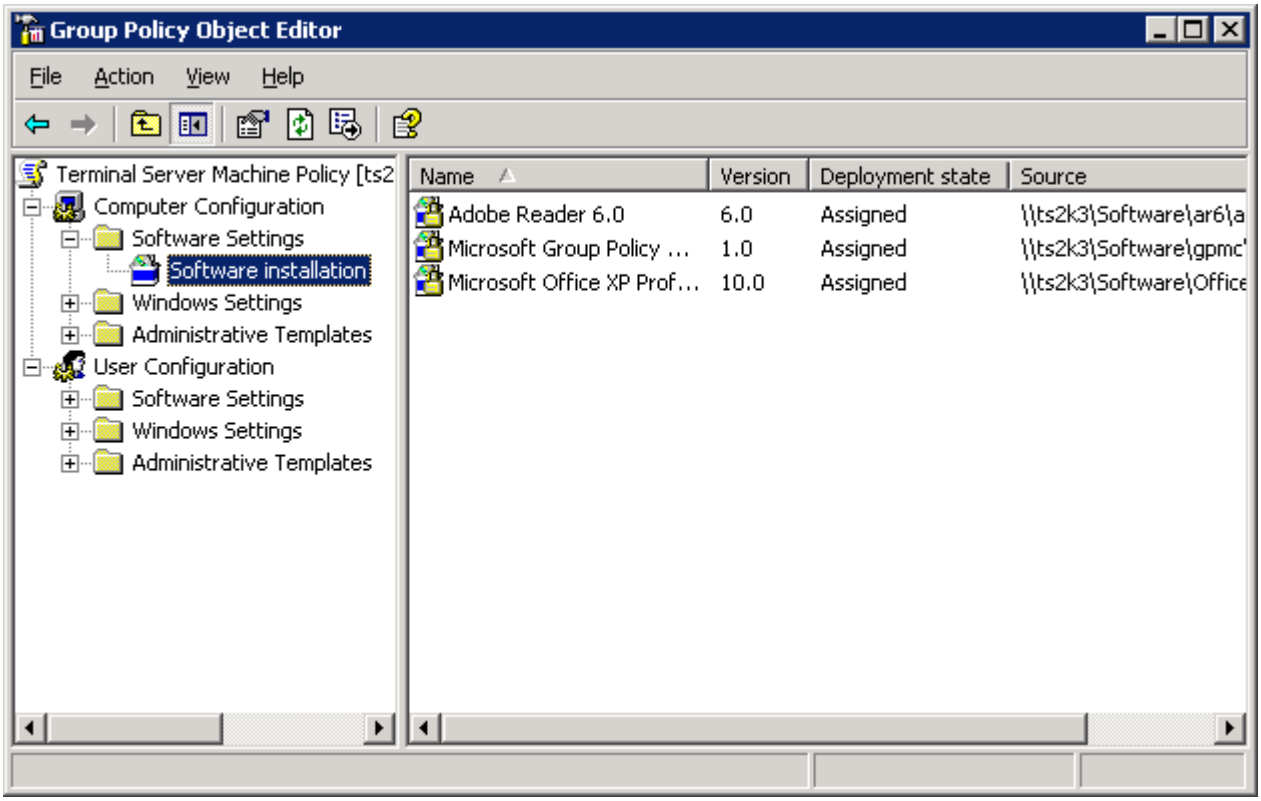


***Figure 2.4: A Group Policy that has three software packages applied.***

Let's assume that you want to install Acrobat Reader and Office XP on all terminal servers to which the GPO applies, but you only want to install the Group Policy Management Console on the terminal servers that administrators use. By default, the packages will inherit security settings from the GPO, so any computers that have read permission on the GPO will also have read permission on the package, and will therefore install the software during boot up.

You can change the permissions on the package and limit the ability to read the package to only a specific group of computers. By doing so, all computers in the organizational unit (OU) will process the policy but only those with read permission on the package will install it. Figure 2.5 shows the default permissions on a package and the permissions after they have been modified so that only the Admin Terminal Servers group can read it.

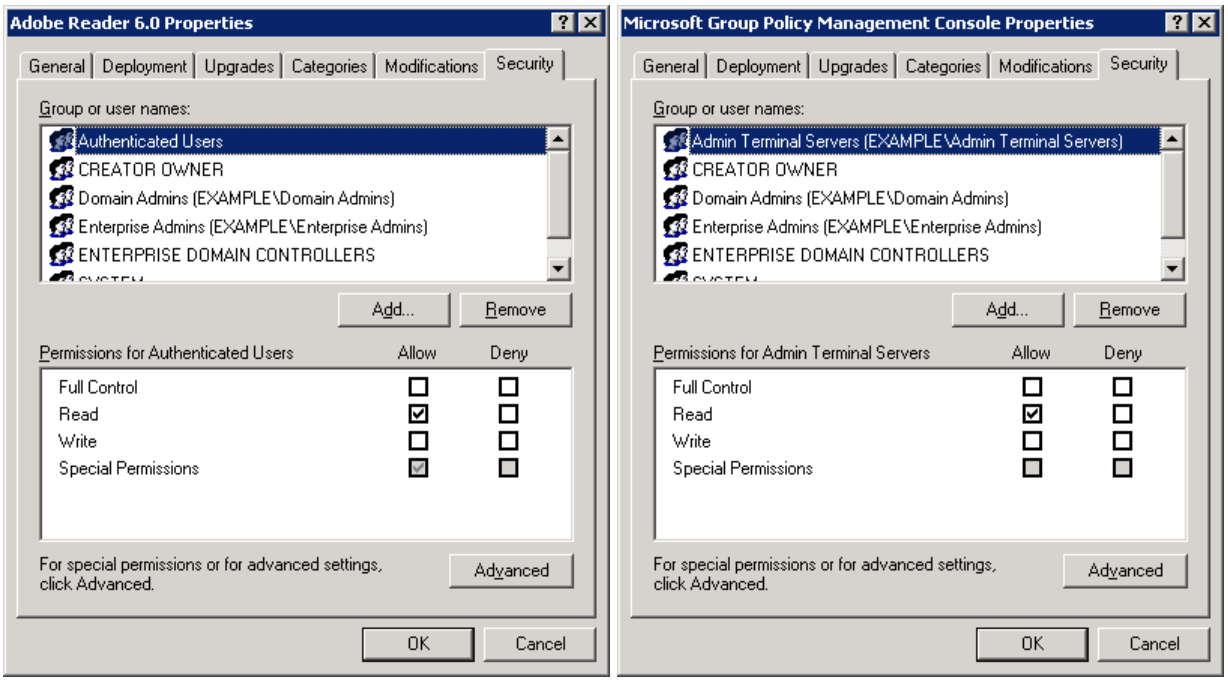


***Figure 2.5: Filtering security on a software package.***

For this security filter to work, you will need to create a Domain Security group—Admin Terminal Servers, for example—and place the servers you want to receive the console into that group.

> ☞ To modify the permissions of a package, you need to break inheritance on the GPO's permissions. To do so, click Advanced, and clear the *Allow inheritable permissions* check box. You can then choose to copy the existing permissions and use them as a starting point for your modifications.

### *MSIEXEC Command-Line Reference*

The following list highlights some common command-line arguments for Windows Installer (msiexec.exe):

- /I—Installs an msi package
- /f—Repairs an installed package
- /a—Creates an Administrative Installation package
- /x—Uninstalls a package
- /l logfile—Creates a log at the specified path/filename
- /lv logfile—Creates a verbose log
- /p—Applies a patch in MSP format
- /q with n|b|r|f—Sets the user interface (UI) level
- /qn—Silent installation
- /qn+—Silent installation with completion dialog box

For example, the command:

```
Msiexec /i pro11.msi /lv c:\temp\office.log /qn+
```

Installs Office 2003 silently with a completion dialog box and logs the installation to c:\temp\office.log.

You can modify the default installation behaviors of an MSI file by using a *transform*—an MST file. You can specify a transform from the command line by using the syntax:

```
Msiexec /i pro11.msi /lv /qn+ transforms=terminalserver.mst
```

> ☞ If the MSI and MST files are not both in the current working directory, you should specify their exact path. UNC paths are recommended so that the Windows Installer service can locate the source files in the event that a repair is required.

Some MSI files also take custom arguments. For example, if you want to install Office 2003 on a terminal server via the MSI package (skipping the setup.exe wrapper), you must add

```
Terminalserver=1
```

to the msiexec command line or to the package properties in the Group Policy.

## Application Compatibility Scripts

During the logon process, all terminal servers call a built-in script called usrlogon.cmd; this script is used to map the root drive (if one is defined) as well as call any application compatibility scripts that have been installed. Between the application compatibility subsystems and the Windows Installer Service, very few applications still require application compatibility scripts. In fact, WS2K3 only comes pre-loaded with one—Eudora 4.

Application capability scripts are used to copy files to a user's home directory or user profile, or to modify registry keys that are not handled via registry mapping. They can also be used to map drives that are required for specific applications.

### *How to Add an Application Compatibility Script that Does Not Require a Root Drive*

If your application compatibility script does not reference the root drive, copy the script to C:\Windows\Application Compatibility Scripts\logon, then add a call statement to the usrlogn1.cmd file located in the Sytem32 directory. If this script is the first application compatibility script you are installing on the server, you will need to create this file. The call statement should look like this:

```
call scriptname.cmd
```

### *How to Add an Application Compatibility Script that Requires a Root Drive*

To call an application compatibility script that requires a root drive, you must first define the root drive letter. To do so, run the CHKROOT.CMD script found at C:\Windows\Application Compatibility Scripts. Once you do so, the root drive variable will be defined during the logon process, and you can reference it in your application compatibility script logon scripts.

Next, copy your application compatibility script logon script to C:\Windows\Application Compatibility Scripts\logon. Finally, go to the system32 directory and add a call statement to the usrlogn2.cmd file. If this script is the first application compatibility script that you are installing on the system, you will need to create this file. The call statement should look like this:

```
call scriptname.cmd
```

Usrlogn2.cmd only gets run if the root drive letter has been defined.

## User Logon Process and Scripts

The logon process to a terminal server can involve several scripts depending on the environment. Scripts can be written in whatever scripting language you prefer; however, natively, WS2K3 only supports Shell Script (BAT and CMD files), Visual Basic Script (VBS Files), and Java Script (JS files). If you want to use another language (KIX or Perl, for example), be sure to deploy the appropriate runtime to your servers. Figure 2.6 offers a flowchart showing the scripts and the order that they are processed.

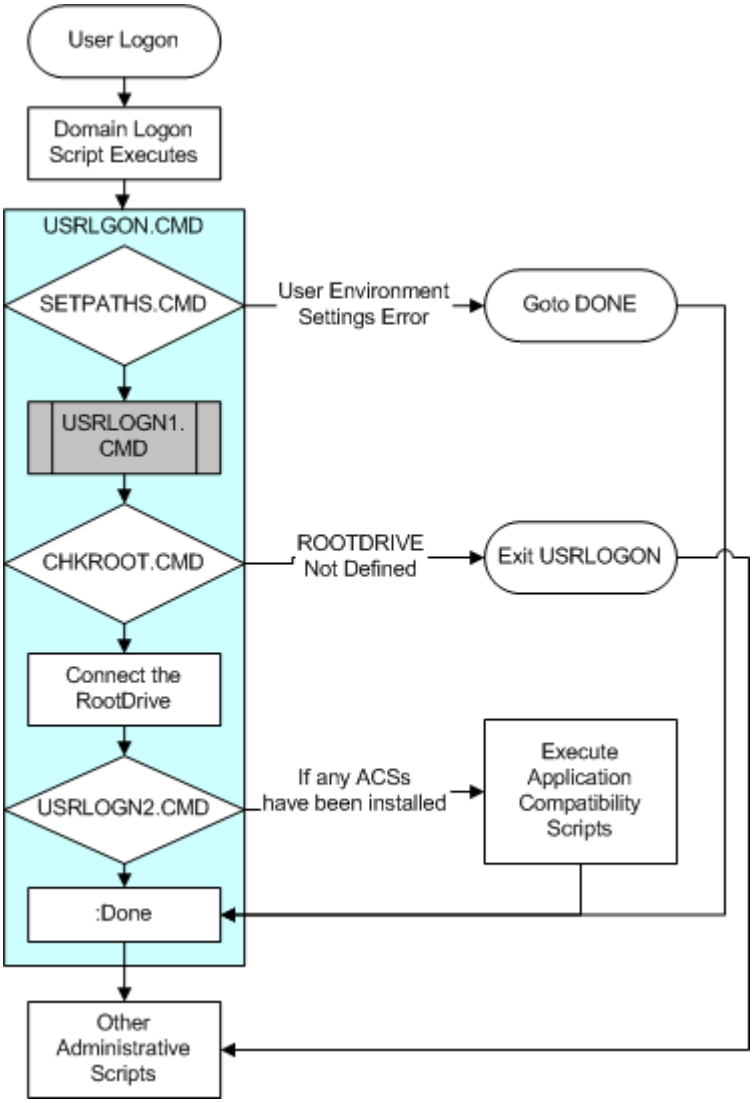


***Figure 2.6: The user logon process, and its associated scripts***

As you can see, there are several options for invoking user logon scripts. You should select the one that is right for your environment. Some things to consider:

- Does the script perform operations specific to a given application? If so, you might want to use an application compatibility script so that the logon script is only run on servers that have the application installed.
- Is the script specific to your terminal servers (and should not be run on workstations)? If so, consider a Group Policy-based logon script.
- Is the script universal? A drive mapping to a public drive for example? Then either a Group Policy or per-user script might be the best option.
- Is the script unique to a given user? If so, then a per-user script is the way to go.

### How to Invoke a Per-User Logon Script

Per-user logon scripts are usually domain based and are configured as an attribute of a user object in AD. To configure a user account to run a logon script:

- Copy the script and any required files (resource kit tools, command-line utilities, and so on that are called by the script) to the domain netlogon share (\\domain\netlogon).
- Launch Active Directory Users and Computers, and open the properties dialog box of a user object (see Figure 2.7).
- On the Profile tab, enter the name of the script in the Logon script field.

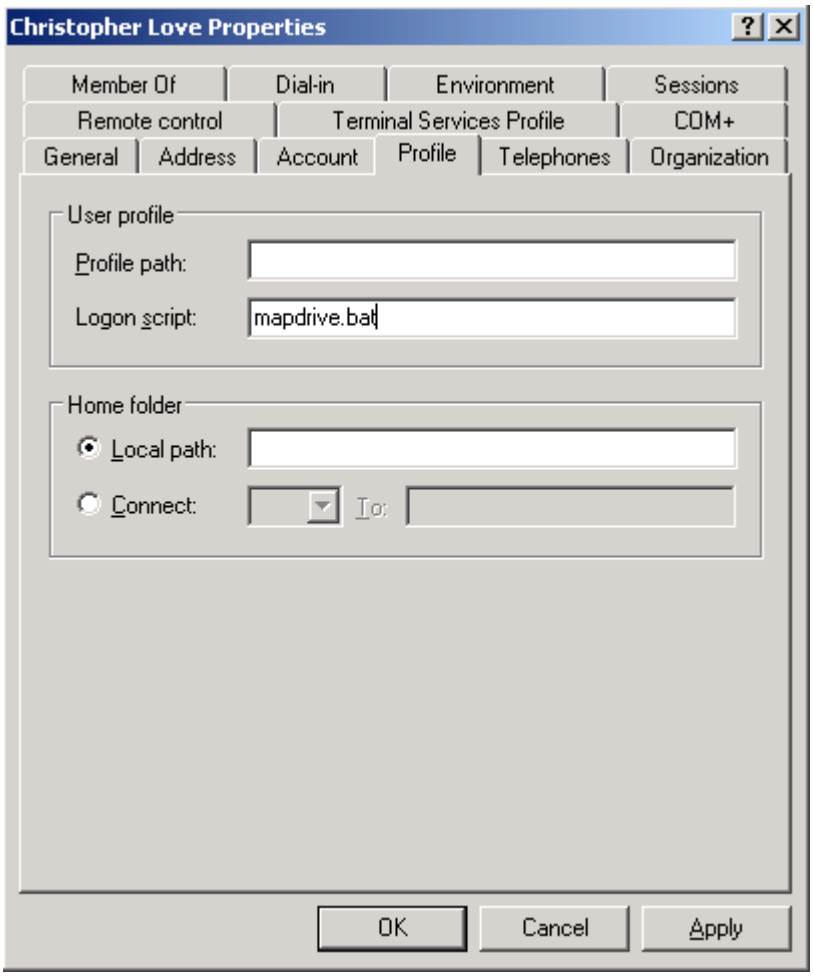


***Figure 2.7: Adding a logon script to a user object.***

### *How to Invoke a Logon Script via Group Policy*

Group Policy-based logon scripts are executed for any user account that is within the scope of the GPO. The following list provides the steps to add a logon script to a GPO (see Figure 2.8):

- Edit a GPO that applies to the desired user objects.
- Drill to User Configuration | Windows Settings | Scripts.
- Double-click Logon in the right pane of the editor.
- In the dialog box that appears, click Show Files.
- Copy the script and any dependant files to the directory that appears.
- Close the Explorer Window, then click Add in the dialog box.
- Click Browse, and select the script.

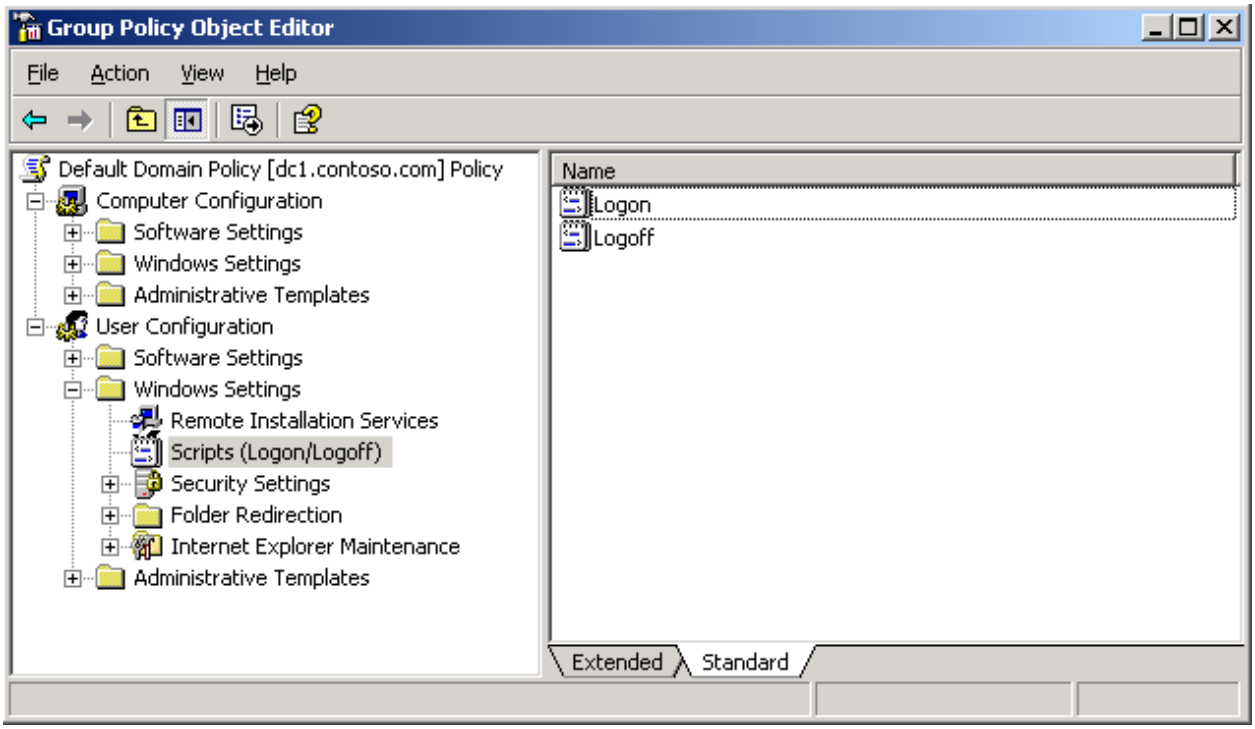


***Figure 2.8: Adding a logon script to a GPO.***

Normally, the GPO you use will be linked to the OU that the user objects are in, but it is common to configure terminal servers to run in Loopback Policy Processing mode, in which case the GPO will be linked to the OU that contains the terminal server computer objects.

> ☞ By default, you use Active Directory Users and Computers to access and edit GPOs. Strongly consider installing and using the Group Policy Management Console to manage your GPOs instead. The console is a free download from Microsoft and can be found at http://go.microsoft.com/fwlink/?linkid=21813. The Group Policy Management Console provides a single interface to display, manage, and edit all GPOs in your forest. It also has built-in tools to check the resultant set of policy (RSOP) as you perform GPO modeling "What if?" scenarios.

## Summary

Although most applications can be installed and run in a terminal server environment without requiring any changes by the administrator, it is useful to have the knowledge for how to deal with applications that necessitate modification. In such cases, you can rely on tools such as application registry mapping, INI file mapping, and root drive specification as well as modified application compatibility scripts and logon scripts. We'll build on this knowledge in the next chapter, which explores user sessions and profiles in a terminal server environment.

## Content Central



## Download Additional eBooks!