realtimepublishers.com™

The Definitive Guide™ To

Windows Server 2003 Terminal Services

triCerat Software®

Gresyon Mitchem

Copyright Statement

Chapter 4: Terminal Services Administration

As with any technology deployment, the process of installing and configuring your terminal server is only half the work. You must also plan for ongoing administration and maintenance and software life cycle management. In this chapter, we'll focus on Terminal Services administration, including user account configuration and management. In addition, we'll explore GPO-based configuration from an AD perspective. I will introduce you to loopback policy processing, the creation of custom administrative templates, and the domain policy processing order. I will also walk you through some common terminal server administrative tasks and introduce you to the tools—both GUI and command line—used to manage terminal servers and user sessions. Let's start by jumping into user account administration.

Terminal Server Access Requirements

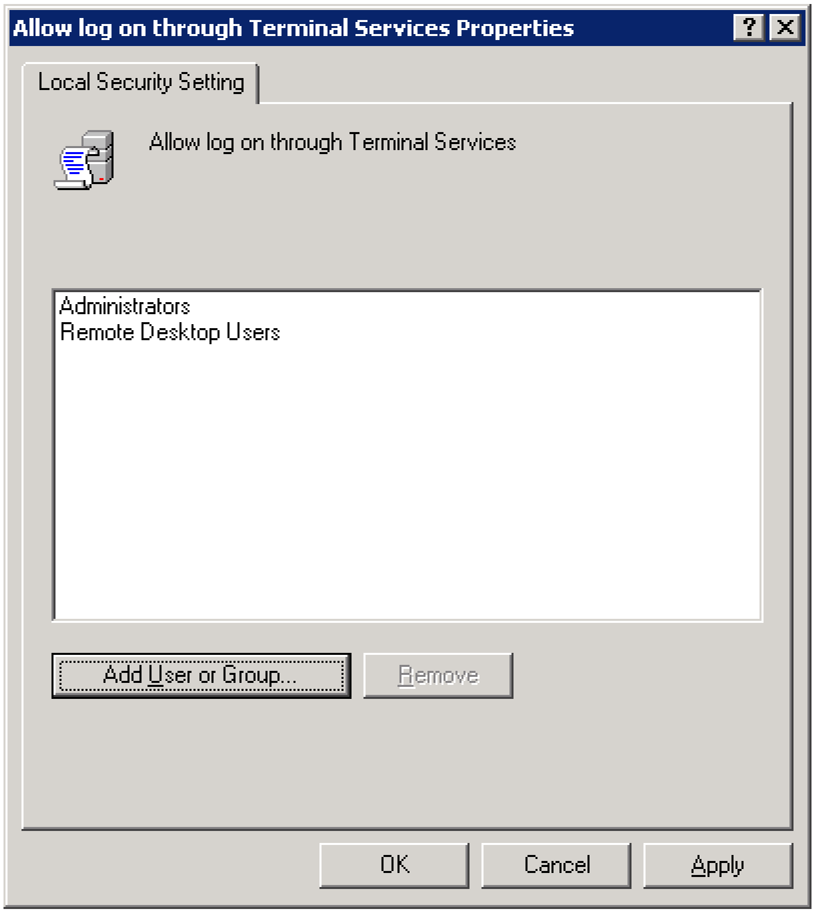
WS2K3 has three distinct layers of protection that enable you to control who can log on to a terminal server. For a user to log on to a terminal server, these settings must be in place:

- The *Allow log on through Terminal Services* right—Under Win2K, you are required to grant the *Log on locally* right to all users who need access to a terminal server. This requirement poses a potential security hole as it allows users to log on at the console of the server, thus bypassing any restrictions you configured for RDP. WS2K3 separates the right to log on to the console from the right to log on through Terminal Services. By default, on WS2K3, the *Allow log on through Terminal Services* right is granted to Administrators and to the Remote Desktop Users group.
- Permission to use RDP—An administrator can set permissions on RDP through the Terminal Services Configuration tool. As I mentioned in Chapter 2, Microsoft's new focus on security has changed the default permissions for the protocol in WS2K3. Under Win2K, the local Users group is granted access to RDP; WS2K3 restricts this right to the local Remote Desktop Users group. Thus, you must add your users to this group in order for them to log on to the terminal server.
- The *Allow logon to terminal server* check box—In the properties of each user object in AD, there is an *Allow logon to terminal server* check box that controls whether the user is enabled to log on to a terminal server. This check box is selected by default.

Two of the three settings are dependent on membership in the Remote Desktop Users group. If a user receives a *You do not have permission to access this session* error message, one of these three settings is the culprit. In the following sections, I will explain how and where to adjust these settings.

Allow Log On Through Terminal Services

You can control the *Allow log on through Terminal Services* right through either the Local Security Policy administrative tool or the GPO editor (GPEDIT.MSC), under Security Settings, Local Policies, User Rights Assignment. When you install the Terminal Services role, this right is granted to the local Remote Desktop Users group. Figure 4.1 shows the default groups that are granted this right.



Allow log on through Terminal Services Properties

Local Security Setting

Allow log on through Terminal Services

Administrators
Remote Desktop Users

Figure 4.1: Groups assigned the* Allow log on through Terminal Services *right by default.

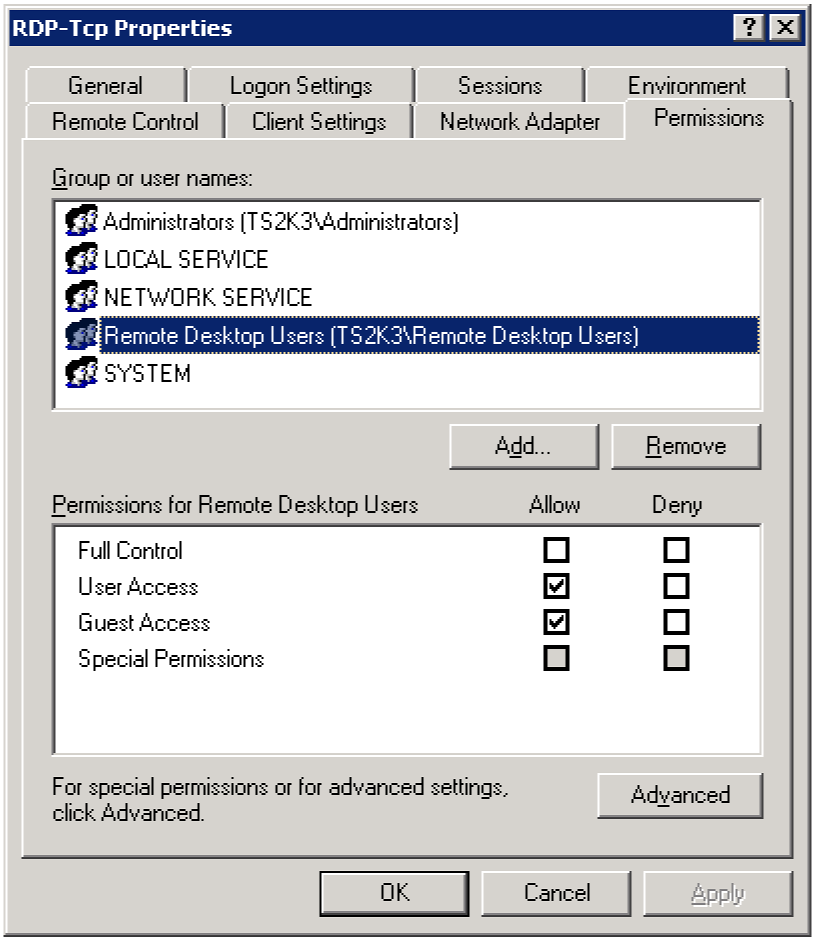
If your terminal server is in an AD domain, the Local Security Policy editor will show both the local setting and the effective setting, as user rights assignment can be controlled through domain GPOs. If you find that the effective setting does not grant the required users this right, you will need to use the Resultant Set of Policy tool (RSOP.MSC) to determine which GPO is revoking the right.

> ☞ Even on WS2K3, the *Log on locally* right is open to both Administrators and users. If your terminal server is in an unsecured location, you can restrict this right to Administrators only, and only allow users to access the server over Terminal Services.

Permissions on RDP

In Chapter 2, I introduced you to the Terminal Services Configuration tool, and focused on how you can use this tool to tune the server for optimal performance and to configure user session timeouts and resource redirection settings. You can also use this tool to set permissions on RDP.

As the Permissions tab of the properties of the RDP-Tcp connection shows (see Figure 4.2), permission to use RDP is restricted to Administrators and the Remote Desktop Users group. By default, the Remote Desktop Users group is empty, so you must add users or groups to it to enable them to connect to your terminal server.



Group or user names:

- Administrators (TS2K3\Administrators)
- LOCAL SERVICE
- NETWORK SERVICE
- Remote Desktop Users (TS2K3\Remote Desktop Users)
- SYSTEM

Permissions for Remote Desktop Users	Allow	Deny
Full Control	☐	☐
User Access	☑	☐
Guest Access	☑	☐
Special Permissions	☐	☐

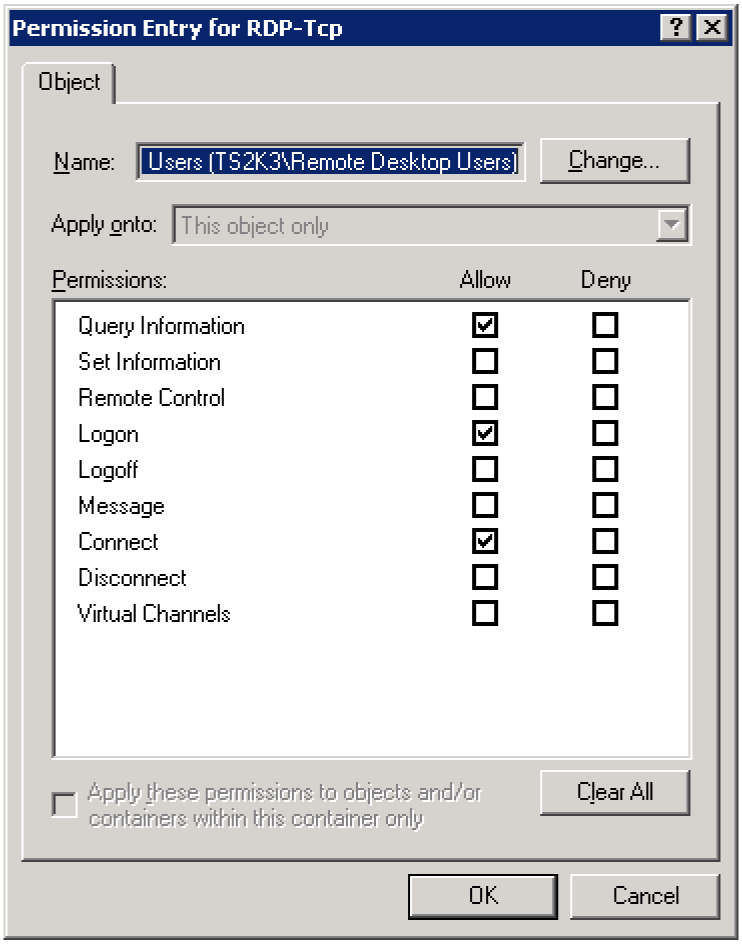
For special permissions or for advanced settings, click Advanced.

Figure 4.2: Setting permissions on RDP.

As you can see in Figure 4.2, the Remote Desktop Users group is granted User Access by default. Each access level—guest, user, and full control—comes with a different set of permissions over sessions on the terminal server. To fully utilize the power of these permissions, you must first understand what each access level provides as well as which advanced settings are available.

RDP Access Levels

RDP provides three basic levels of access: Guest Access, User Access, and Full Control. The level assigned to a group determines the group's abilities when connected to the terminal server over RDP. Let's first examine the permissions available, then associate them with the basic access levels. Figure 4.3 shows the advanced ACL editor's list of individual permissions, and Table 4.1 explains which abilities each permission setting bestows on the users.



Permission Entry for RDP-Tcp

Object

Name: Users (TS2K3\Remote Desktop Users)

Apply onto: This object only

Permissions:	Allow	Deny
Query Information	☑	□
Set Information	□	□
Remote Control	□	□
Logon	☑	□
Logoff	□	□
Message	□	□
Connect	☑	□
Disconnect	□	□
Virtual Channels	□	□

□ Apply these permissions to objects and/or containers within this container only

Figure 4.3: Permissions available for RDP.

Permission	**Ability**
Query Information	Query information through the Terminal Services Administrator or at a command prompt using the QUERY command.
Set Information	Change settings and permissions for RDP.
Remote Control	View or actively control another user's session.
Logon	Log on to a session on the server.
Logoff	Force another user to logoff of his or her session.
Message	Send messages to other sessions on the server by using the Terminal Services Manager console or at a command prompt by using the MSG command.
Connect	Reconnect to a session that the same user left active on the server.
Disconnect	Forcibly disconnect another user from his or her session, leaving the session active on the server.
Virtual Channels	Use virtual channels, which are communication channels that developers can use to enhance the capabilities of RDP. As long as System has this permission, users will be able to use applications that take advantage of virtual channels.

Table 4.1: Permissions available for RDP.

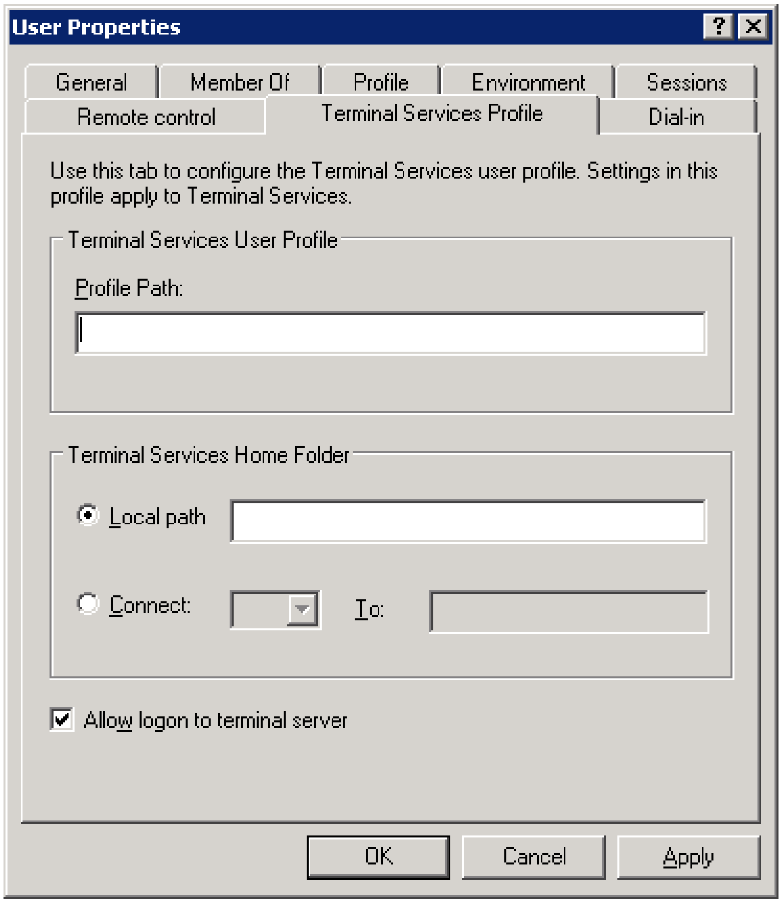
Table 4.2 shows the permissions assigned to each basic access level. You can use the advanced window of the ACL editor to create special permission sets. For example, you may want your Help desk staff to have all the abilities of Full Control except Set Information.

Permission	Guest Access	User Access	Full Control
Query Information		X	X
Set Information			X
Remote Control			X
Logon	X	X	X
Logoff			X
Message			X
Connect		X	X
Disconnect			X
Virtual Channels			X

Table 4.2: Permissions associated with the basic access levels.

Allow Logon to Terminal Server

The last requirement to log on to a terminal server is a per-user setting. In the properties of every user object, there are four tabs dedicated to terminal server settings. Most of the settings affect session behavior when a user connects to a terminal server. I will cover these settings in the next section. However, you can use the *Allow logon to terminal server* setting to restrict a user's ability to connect to your terminal servers altogether. Figure 4.4 shows this setting on the Terminal Services Profile tab of the User Properties. This setting is enabled for all users by default.



Use this tab to configure the Terminal Services user profile. Settings in this profile apply to Terminal Services.

Terminal Services User Profile

Profile Path:

Terminal Services Home Folder

● Local path

○ Connect: To:

☑ Allow logon to terminal server

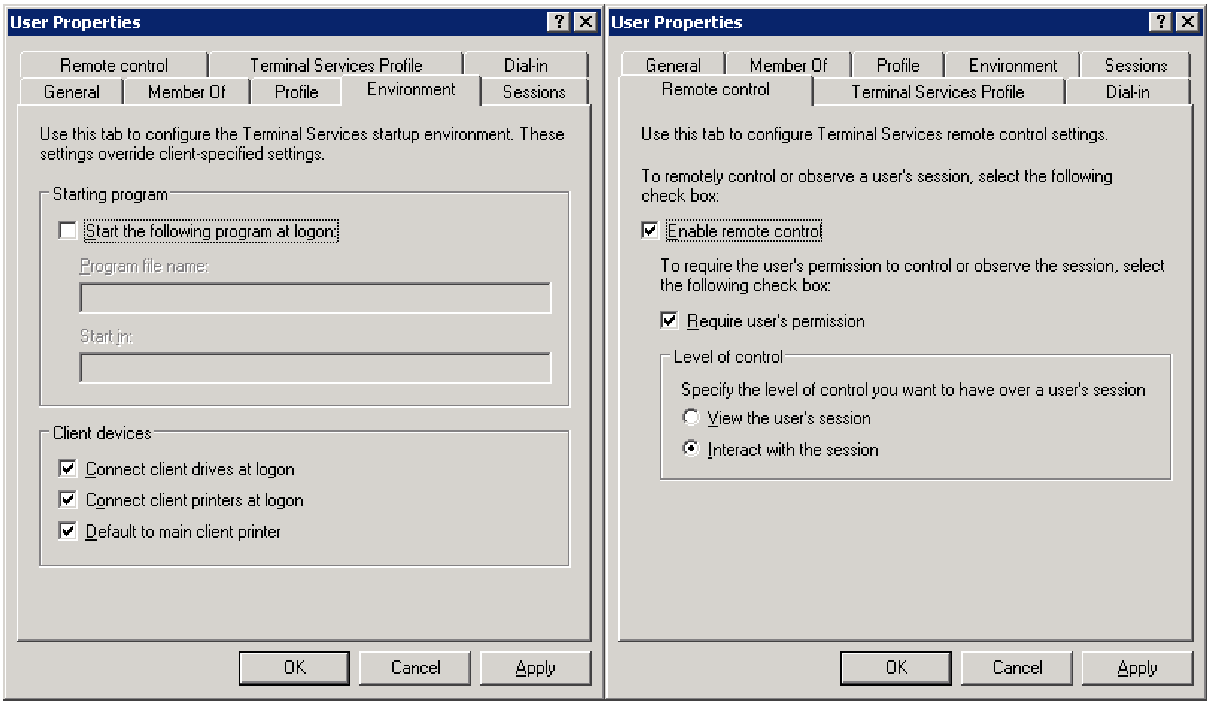
Figure 4.4: The Terminal Services Profile tab of a user object.

User Account Configuration

While we are focused on the User Properties interface, I'd like to take you through the rest of the terminal server settings. Most of these settings are also available in the Terminal Services Configuration tool. It is up to you to decide whether you want to control them on a per-server or a per-user basis. Keep in mind that per-server settings override per-user settings. To access the user-based options, use one of the following tools: for AD domains, use the Active Directory Users and Computers tool; for NT 4.0 domains, use the User Manager for Domains tool; and for Workgroup mode terminal servers, use the Computer Management Administration tool.

> ✎ The Terminal Services tabs shown in this section do not appear in the NT 4.0 version of User Manager for Domains; you must use the Win2K, WS2K3, or NT 4.0 Terminal Server Edition version of the tool.

Regardless of the tool you use, the same options are available. Figure 4.5 shows the Environment and Remote control tabs of the User Properties interface.



User Properties

Remote control | Terminal Services Profile | Dial-in
General | Member Of | Profile | Environment | Sessions

Use this tab to configure the Terminal Services startup environment. These settings override client-specified settings.

Starting program

□ Start the following program at logon:

Program file name:

Start in:

Client devices

☑ Connect client drives at logon

☑ Connect client printers at logon

☑ Default to main client printer

OK | Cancel | Apply

User Properties

General | Member Of | Profile | Environment | Sessions
Remote control | Terminal Services Profile | Dial-in

Use this tab to configure Terminal Services remote control settings.

To remotely control or observe a user's session, select the following check box:

☑ Enable remote control

To require the user's permission to control or observe the session, select the following check box:

☑ Require user's permission

Level of control

Specify the level of control you want to have over a user's session

○ View the user's session

● Interact with the session

OK | Cancel | Apply

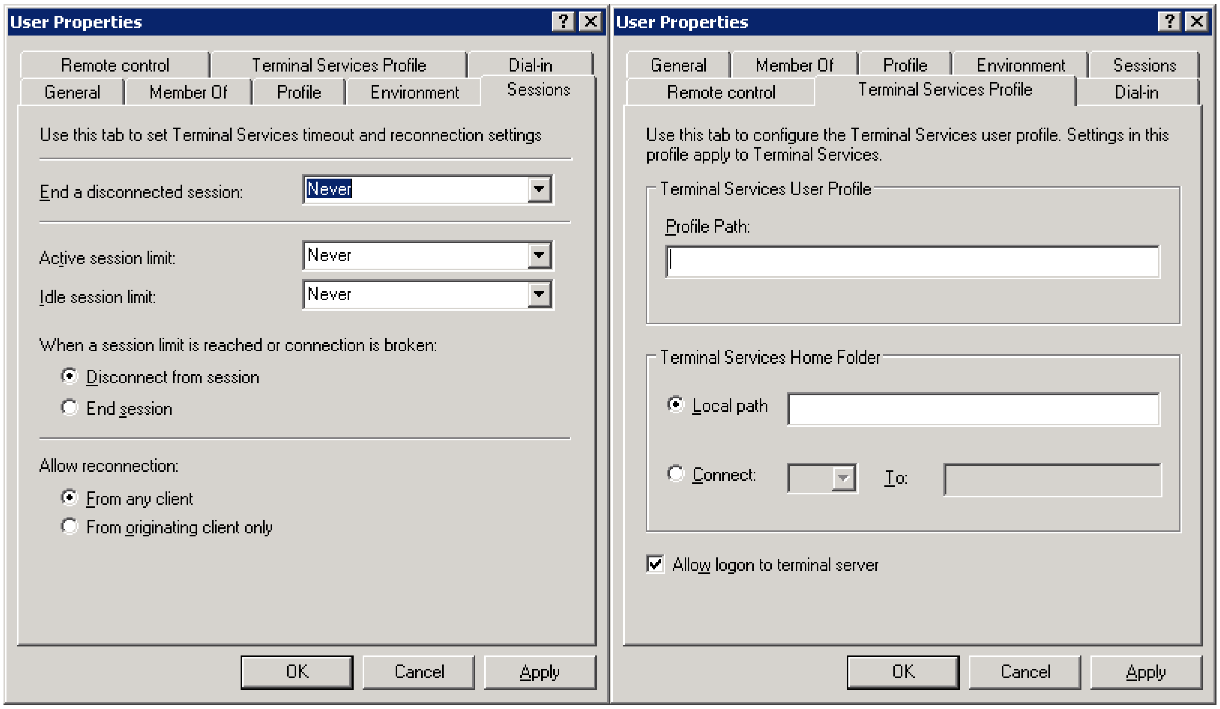
Figure 4.5: The Environment and Remote control tabs of the User Properties interface.

The Environment tab is used to configure both starting program and client device resource redirection settings. You use the client device settings to enable or disable the automatic connection to the client devices' drives and printers. If you enable the *Start the following program at logon* setting, whenever the user connects to any terminal server, he or she will receive the specified program instead of a Windows desktop.

As with all of the settings in this section, server settings override user settings. Thus, if you specify a starting program on the user account as well as one in the Terminal Services Configuration tool, the program specified on the server will be launched.

On the Remote control tab, you can enable or disable the ability to remotely control sessions belonging to this user. If you enable remote control, you can also specify whether to require the user's permission before the remote control connection is permitted as well as the level of control the administrator has over the session once connected. Once again, these settings will be overridden if remote control is configured on the server through the Terminal Services Configuration tool.

The Sessions tab, which Figure 4.6 shows, allows you to set timeout values for the user's Terminal Services sessions. On this tab, you can set timeouts for active, idle, and disconnected sessions. You can specify whether to immediately end the session when the connection is lost or the active session time limit is reached or to treat the session as disconnected. You can also specify whether the user can reconnect from any client device or only from the device that originally started the session.



User Properties

Use this tab to set Terminal Services timeout and reconnection settings

End a disconnected session: Never

Active session limit: Never

Idle session limit: Never

When a session limit is reached or connection is broken:

● Disconnect from session

○ End session

Allow reconnection:

● From any client

○ From originating client only

User Properties

Use this tab to configure the Terminal Services user profile. Settings in this profile apply to Terminal Services.

Terminal Services User Profile

Profile Path:

Terminal Services Home Folder

● Local path

○ Connect: To:

☑ Allow logon to terminal server

Figure 4.6: The Sessions and Terminal Services Profile tabs of the User Properties interface.

You use the Terminal Services Profile tab, which Figure 4.6 also shows, to set the profile and home directory path to be used when the user logs on to a Terminal Services session. In addition, this tab provides the settings to specify whether the user can log on to a terminal server at all. Unlike the other settings in this section, the settings on this tab are not duplicated in the Terminal Services Configuration tool.

Home and Profile Directories

As a systems administrator, you are probably familiar with network home directories and roaming profiles. These features in Windows enable you to maintain central stores for your users' documents and profile settings so that the documents and profile settings are available regardless of at which computer users sit.

Terminal Services has the ability to maintain separate stores for home and profile data for the users. How you utilize this ability depends on your environment and administrative style.

Terminal Services Profile Path

When a user logs on to a workstation, the system checks the Profile Path attribute of the user object to see whether the user has a centrally stored profile. If so and if it is newer than any locally cached copy, the profile is downloaded for the user. In the same way, when a user logs on to a terminal server, the system queries the UserParameters attribute and looks for a Terminal Services Profile Path.

This separation enables administrators to maintain separate profiles for users depending on which type of computer the users are accessing. In most cases, you will want to take advantage of Terminal Services profiles, as certain functions of Terminal Services make it difficult to not maintain Terminal Services profiles. Let me explain what I mean. If you do not use roaming profiles for your users' workstations, you rely on the fact that the computer maintains a copy of the user profile. If your users do not log on to more than one PC, this setup is perfectly fine. On a terminal server, however, not using roaming profiles means that the terminal server would be maintaining the profiles for *all* your users, which would consume a lot of disk space. Also, if you need to scale your Terminal Services infrastructure and use load balancing and Session Directory to distribute your users among more that one terminal server (and you were not taking advantage of Terminal Services profiles), your users would be maintaining separate profiles on each server.

Implementing Terminal Services profiles addresses both of these issues. Having a central Terminal Services profile enables the user to receive the same settings regardless of to which server the Session Directory connects the user. To alleviate the disk space problem, you can enable a System Policy that deletes cached copies of roaming profiles. This way, after a user logs off of the terminal server, the disk space is reclaimed once the system copies the profile back to the central location.

> If you do not define a Terminal Services profile path but define a Windows roaming profile path, the terminal server will use the Windows profile. Also, if the Terminal Services profile is defined but unavailable, the system will fall back on the Windows profile. This behavior may have undesired results if you are using application compatibility scripts on your server.

If you use roaming Windows profiles, using Terminal Services profiles can be even more critical, because if the system does not find a Terminal Services profile path in the user's account, it will then look for a Windows profile path and use it instead. In Chapter 5, you will learn that some applications require application compatibility scripts to function properly on terminal servers. These scripts often make changes to registry settings under the HKEY_CURRENT_USER hive to help tune applications for simultaneous users. If these changes are made to the user's Windows profile, the user might experience problems the next time he or she logs on to a workstation.

Geography may also be a factor in separating your profiles. You probably store your users' Windows profiles on file servers that are close to their workstations, but your terminal servers, given their low-bandwidth requirements, may be in a central data center. You do not want users to have to pull their profiles across a slow WAN link.

> WS2K3 has two new Group Policy settings that control user profiles. The *Allow only local user profiles* setting prevents a specific computer from downloading roaming profiles even if one is configured on the user account. The *Set Path for TS Roaming Profiles* setting allows you to configure a specific file server to be used for roaming profiles for all users logging on to a terminal server.

Terminal Services Home Directories

You are also able to configure your user accounts to use separate home directories when logging on to a terminal server. As you will learn in Chapter 5, the system uses the user's home directory as its ROOTDRIVE and stores application compatibility files there. Microsoft designed the ability to use a separate home directory when logging on to a terminal server to keep these files out of the user's Windows home directory. The problem is that if your users store their documents in their Windows home directory, the users will need that same directory available when logging on to a terminal server. If you define a Terminal Services home path, that path will be mapped instead of the Windows home path during logon. If you do not define a Terminal Services home path, the Windows home path is mapped as usual.

Most users will not mind the few files and directories that application compatibility scripts create and will simply ignore them. If you want, you can modify your application compatibility scripts to flag these directories as hidden to prevent them from bothering your users.

> ☞ As with profiles, if you do not define a Terminal Services home path, the system will use the Windows home path instead. Thus, if you want to use the same home directory for both workstations and terminal servers, simply leave the Terminal Services home path blank.

Configuring User Properties Through the Active Directory Service Interfaces

Using the GUI tools for user account configuration is fine for managing a small group of users or making one-off changes, but if you want to configure a large number of accounts, you may find it easier to use the Active Directory Service Interfaces (ADSI). This feature is a great improvement over Win2K, for which you had to be a skilled C programmer to access these attributes.

You access ADSI by using the Windows Script Host (WSH), thus you can choose whether to write your scripts in Visual Basic Script (VBScript) or Java Script. The examples that I provide will be in VBScript.

Configuring user properties through ADSI is a three-step process. First, you must open a connection to the user account, then set the properties, and finally write the changes back to the user account. To open a connection to the user account, you will use either the WinNT provider or the Lightweight Directory Access Protocol (LDAP). The WinNT provider is used for Security Accounts Manager (SAM) accounts—either local accounts on the terminal server or user accounts in an NT 4.0 domain—and LDAP is used for AD accounts.

> Although you can use the scripts in this section to configure both NT 4.0 domain and Win2K AD accounts, you can only run the scripts on a WS2K3 server; they will not work if run on Win2K or even Windows XP.

The syntax for the connection is either

```
      Set objUser = GetObject("WinNT://<domain name>/<username>,user"
```

or

```
      Set obUser = Get Object("LDAP://<distinguished name of user>")
```

As you can see, to use LDAP, you must know the distinguished name of the user object (for example, cn=joe.user,ou=users,dc=example,dc=domain,dc=com), which is difficult if your users are spread across multiple organizational units (OUs). To make it easier, Microsoft enables the ability to use the WinNT provider for AD accounts as well. The domain controller will automatically translate the WinNT call into an LDAP call for you.

Once you have the user account open, you set the parameters that you want to change. The names of the Terminal Services parameters and the syntax for setting them are provided in Listing 4.1.

```
objUser.ConnectClientDrivesAtLogon = [1,0]
objUser.ConnectClientPrintersAtLogon = [1,0]
objUser.DefaultToMainPrinter = [1,0]
objUser.TerminalServicesInitialProgram = ["path to program"]
objUser.TerminalServicesWorkDirectory = ["path to directory"]
objUser.TerminalServicesProfilePath = ["path to directory"]
objUser.TerminalServicesHomeDirectory = ["path to directory"]
objUser.TerminalServicesHomeDrive = ["drive letter:"]
objUser.AllowLogon = [1,0]
objUser.MaxDisconnectionTime = [minutes, 0 for never]
objUser.MaxConnectionTime = [minutes, 0 for never]
objUser.MaxIdleTime = [minutes, 0 for never]
objUser.BrokenConnectionAction = [1,0]
      1 = end session, 0 = disconnect the sesion
objUser.ReconnectionAction = [1.,0]
      1 = original client only, 0 = any client
objUser.EnableRemoteControl = [0,1,2,3,4]
0 = Disable Remote Control
1 = Enable Notify & Enable Interact
2 = Disable Notify & Enable Interact
3 = Enable Notify & Disable Interact
4 = Disable Notify & Disable Interact
```

Listing 4.1: Names and syntax for setting Terminal Services parameters.

Finally, you must write the changed attributes back to the user account:

```
      objUser.SetInfo
```

Now let’s put it together and set all of the terminal server properties for a single user account. Listing 4.2 shows an example.

```
Set objUser = GetObject _
      (“LDAP://cn=joe.user,ou=users,dc=example,dc=domain,dc=com”)
‘ or: Set objUser = GetObject(“WinNT://example/joe.user,user”)
objUser.ConnectClientDrivesAtLogon = 1
objUser.ConnectClientPrintersAtLogon = 1
objUser.DefaultToMainPrinter = 1
objUser.TerminalServicesInitialProgram = “C:\windows\notepad.exe”
objUser.TerminalServicesWorkDirectory = “c:\windows
objUser.TerminalServicesProfilePath = _
“\\server\tsprofiles\joe.user”
objUser.TerminalServicesHomeDirectory = _
“\\server\home\joe.user”
objUser.TerminalServicesHomeDrive = “H:”
objUser.AllowLogon = 1
objUser.MaxDisconnectionTime = 15
objUser.MaxConnectionTime = 0
objUser.MaxIdleTime = 180
objUser.BrokenConnectionAction = 0
objUser.ReconnectionAction = 0
objUser.EnableRemoteControl = 1
objUser.SetInfo
```

Listing 4.2: A script to configure terminal server user properties.

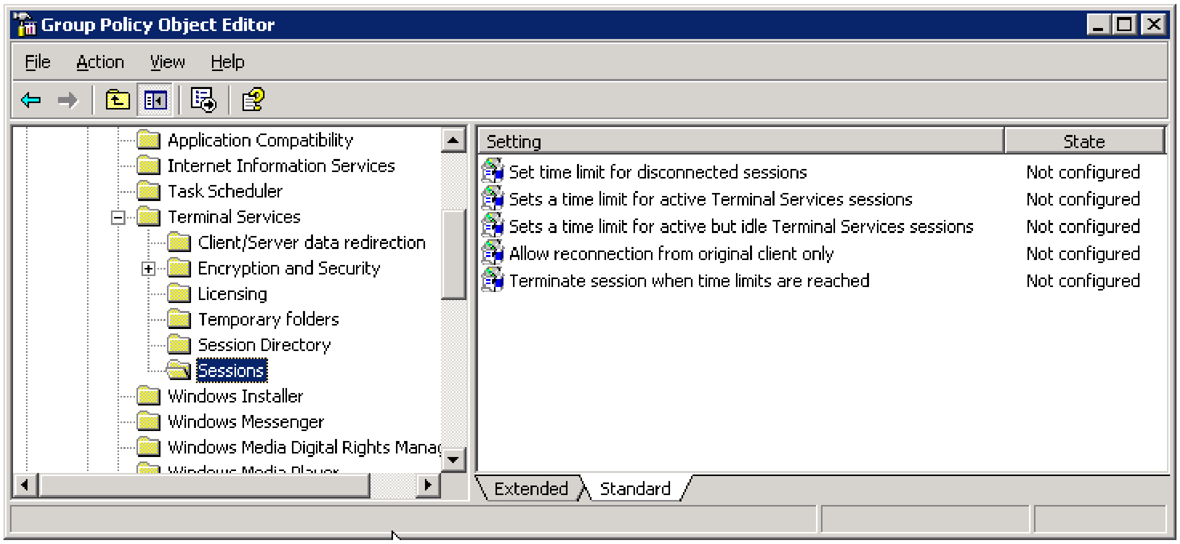
Obviously, if you want to configure a single user account, it would be faster to just use the GUI tool, but ADSI is a great way to configure properties for multiple users at the same time.

> The Microsoft TechNet Script Center (http://www.microsoft.com/technet/scriptcenter) is a great resource for administrative scripting. With a little scripting know-how, you can modify the example scripts to meet your unique needs.

Group Policy Overrides of User Settings

WS2K3 includes a new layer of flexibility when configuring user sessions—Group Policy. As you already know, you can configure session timeouts, client resource settings, and reconnection options on individual user accounts or on specific servers. With WS2K3 Group Policy, you can also choose to control all of these options through GPOs.

To provide even greater flexibility, Microsoft allows you to configure these settings as per-user or per-machine settings within the GPO. Thus, you now have the ability to set all options from a central location for all servers. You can even apply GPOs to specific security groups so that Administrators automatically have access to different options than regular users; you won’t need to go through the trouble of configuring these settings on the individual user accounts. Figure 4.7 shows the Sessions node of the Group Policy Object Editor under Computer (or User) Configuration, Administrative Templates, Windows Components, Terminal Services.


Group Policy Object Editor

Setting	State
Set time limit for disconnected sessions	Not configured
Sets a time limit for active Terminal Services sessions	Not configured
Sets a time limit for active but idle Terminal Services sessions	Not configured
Allow reconnection from original client only	Not configured
Terminate session when time limits are reached	Not configured

Figure 4.7: Configuring session timeouts via GPOs.

All of these options for configuring user and session parameters may seem a little confusing, but once you start using them, they become very familiar. Also, if you are in an AD environment, you will probably find that you stop using the user account attributes and the Terminal Services Configuration tool settings altogether and centralize your settings into GPOs. You should still be aware of the options available in all the tools, as situations will arise that require a unique solution.

If you find yourself needing to set options in multiple locations or you are migrating from an existing Win2K terminal server infrastructure where settings are already configured on your users or servers, you need to be aware of the priority order of the options:

1. Computer Configuration Group Policy settings
2. User Configuration Group Policy settings
3. Terminal Services configuration tool settings
4. User account settings

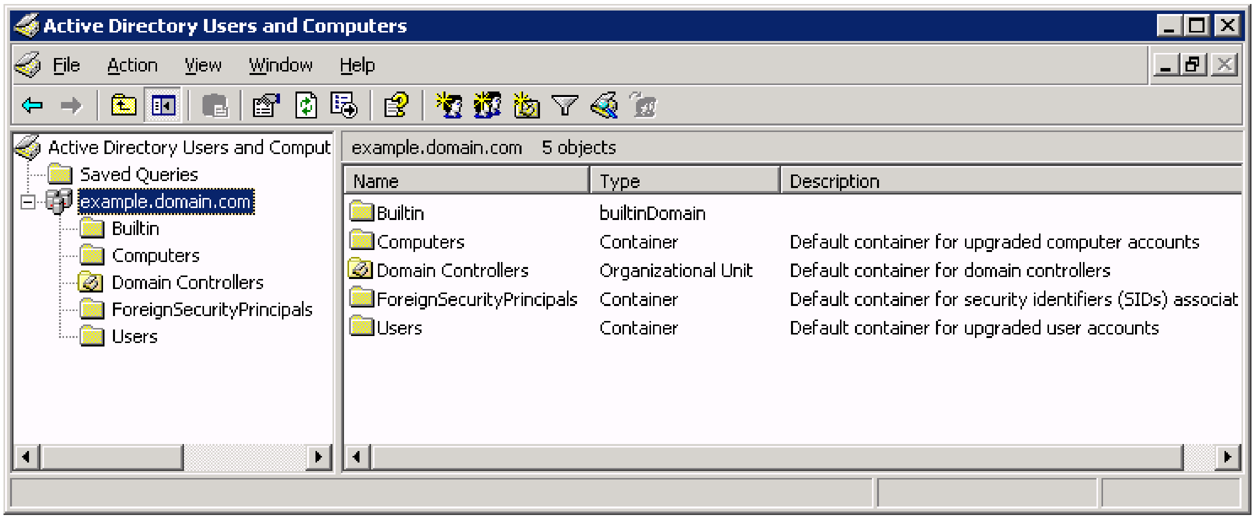
Settings with the highest priority will win, so settings in the Computer Configuration will override User Configuration settings, and so on. As you move to a GPO-based configuration, you will also need to become familiar with Group Policy processing order, which I will cover in the next section.

Managing Terminal Servers in an AD Environment

When working in an AD environment, you have the ability to centralize your terminal server configuration, making it virtually seamless to integrate new servers into your farm. This section will focus on the tools used to configure and manage terminal servers within AD and introduce you to AD Group Policy processing.

Active Directory Users and Computers

Active Directory Users and Computers is the administrative tool used to manage OUs, users, computers, and groups in AD. Figure 4.8 shows the Active Directory Users and Computers interface. From this interface, you have the ability to not only manage the directory and the objects it contains but also quickly access the computer management console for computers in your domain.


Active Directory Users and Computers
File
Action
View
Window
Help
Active Directory Users and Comput
Saved Queries
example.domain.com
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Users
example.domain.com 5 objects

Name	Type	Description
Builtin	builtinDomain	
Computers	Container	Default container for upgraded computer accounts
Domain Controllers	Organizational Unit	Default container for domain controllers
ForeignSecurityPrincipals	Container	Default container for security identifiers (SIDs) associat
Users	Container	Default container for upgraded user accounts

Figure 4.8: The Active Directory Users and Computers interface.

The default configuration for AD organizes the objects into five primary containers:

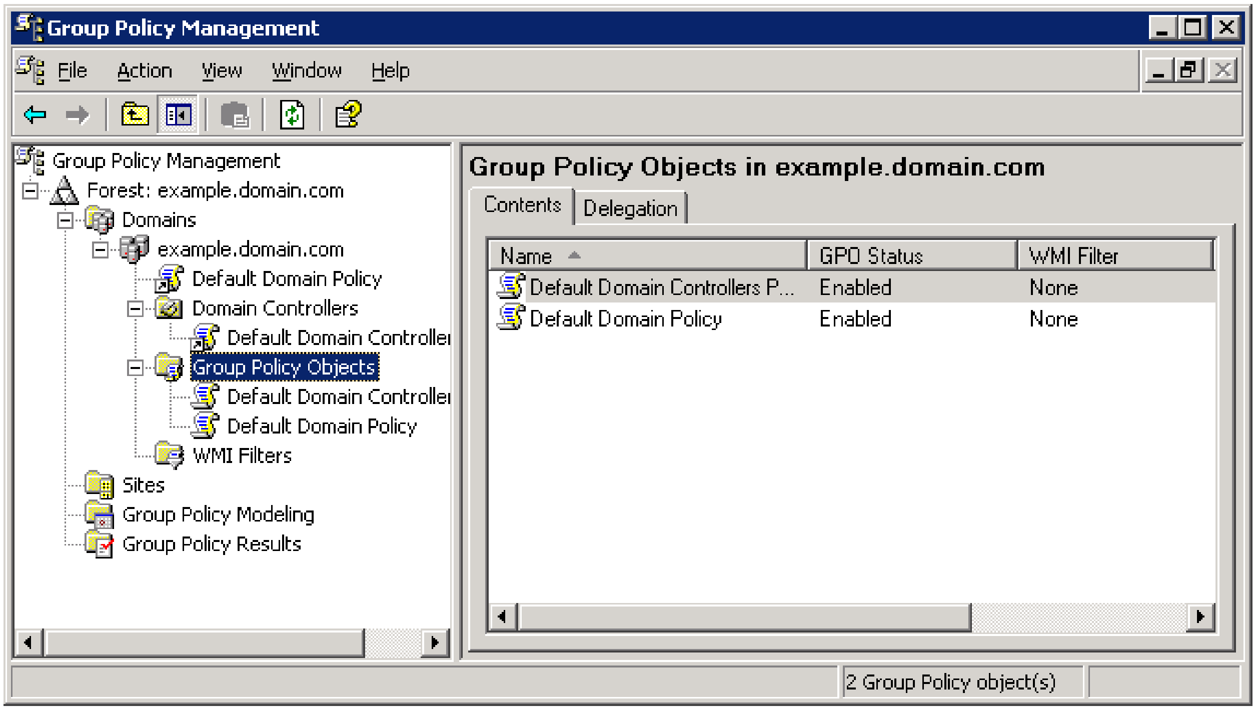
- Users—The default container for user objects, the User container includes built-in users such as Administrator and Guest as well as any users that were already in the domain when upgraded from NT 4.0. This container also holds a number of default domain global groups such as Domain Admins and Domain Users.
- Computers—The default container for computer objects, the Computers container includes any workstations and member servers that were already in the domain when you upgraded from NT 4.0.
- Domain Controllers—The default container for domain controllers.
- Builtin—This container holds the built-in domain local groups that the system uses to manage native rights such as Server and Account Operators and Administrators.
- ForeignSecurityPrincipals—This container is used by the system to hold referenced users and computers from other trusted domains.

If you are managing a small to midsized domain, the default containers might meet your needs. If, however, you need to organize your users and computers for organizational or management purposes, you can create new OU containers to hold the objects. You can then apply permissions to the OUs so that groups of administrators have control only over the users and computers in their own OUs.

When integrating terminal servers into AD, you will most likely want to create a new OU specifically for terminal servers. Doing so gives you the ability to apply GPOs to all servers in the OU without having to worry about adding the servers to security groups in order for them to receive the policy settings. An exception to this suggestion applies if you are exclusively using thin clients for your users; in this case, you will not need to create separate GPOs for workstations and terminal servers.

Group Policy Management Console

With the release of WS2K3, Microsoft also released a new tool for managing GPOs—the Group Policy Management Console (GPMC). This tool is a vast improvement over the GPO management capabilities built-in to Active Directory Users and Computers. Using the GPMC, which Figure 4.9 shows, you can create and edit GPOs; link GPOs to OUs, sites, and domains; configure security; delegate administration; back up and restore GPOs; create HTML-based reports to document settings; and even run Resultant Set of Policy scenarios to help you plan your GPO design.



Group Policy Objects in example.domain.com

Name	GPO Status	WMI Filter
Default Domain Controllers P...	Enabled	None
Default Domain Policy	Enabled	None

Figure 4.9: The GPMC interface.

The GPMC is not installed on WS2K3 by default. To download it and read more about its features and benefits, visit http://www.microsoft.com/windowsserver2003/gpmc/default.mspx.

Configuring Terminal Servers with GPOs

Once you have AD set up and your terminal servers added to the domain, you are ready to start designing your GPOs to configure the servers and the user sessions on them. In Chapter 2, I introduced you to the GPO settings that are used to configure the servers—Session Directory settings, Terminal Services licensing settings, and so on. In the previous section, I mentioned that you can use GPOs to configure user and session settings such as timeouts, resource redirection, and so on. You can also use GPOs to manage the Windows user interface (UI) and to restrict access to tools and features that a malicious user might take advantage of to destabilize your server.

UI Settings

The majority of the settings available in a GPO are dedicated to configuring and locking down the Windows UI. This topic is very volatile, as many users are accustomed to using features that systems administrators often remove or disable in a Terminal Services environment (such as the command prompt, Run command, Task Manager, and so on). I recommend reading the Microsoft article "How to Lock Down a Win2K Terminal Server Session" before you begin this lockdown process. You can then add or remove restrictions based on your users' needs and behavior patterns.

> ✎ Many of the policy settings are designed to eliminate confusion more than to secure the server. For example, you will typically remove the Shut Down command from the Start menu even though non-administrators do not have the rights required to shut down the server.

The following list highlights the policy settings that Microsoft recommends (the settings are divided into Computer Configuration settings and User Configuration settings):

Computer Configuration settings:

- Do not display last user name in logon screen
- Restrict CD-ROM access to locally logged-on user only
- Restrict floppy access to locally logged-on user only
- Disable Windows Installer—Always

User Configuration settings:

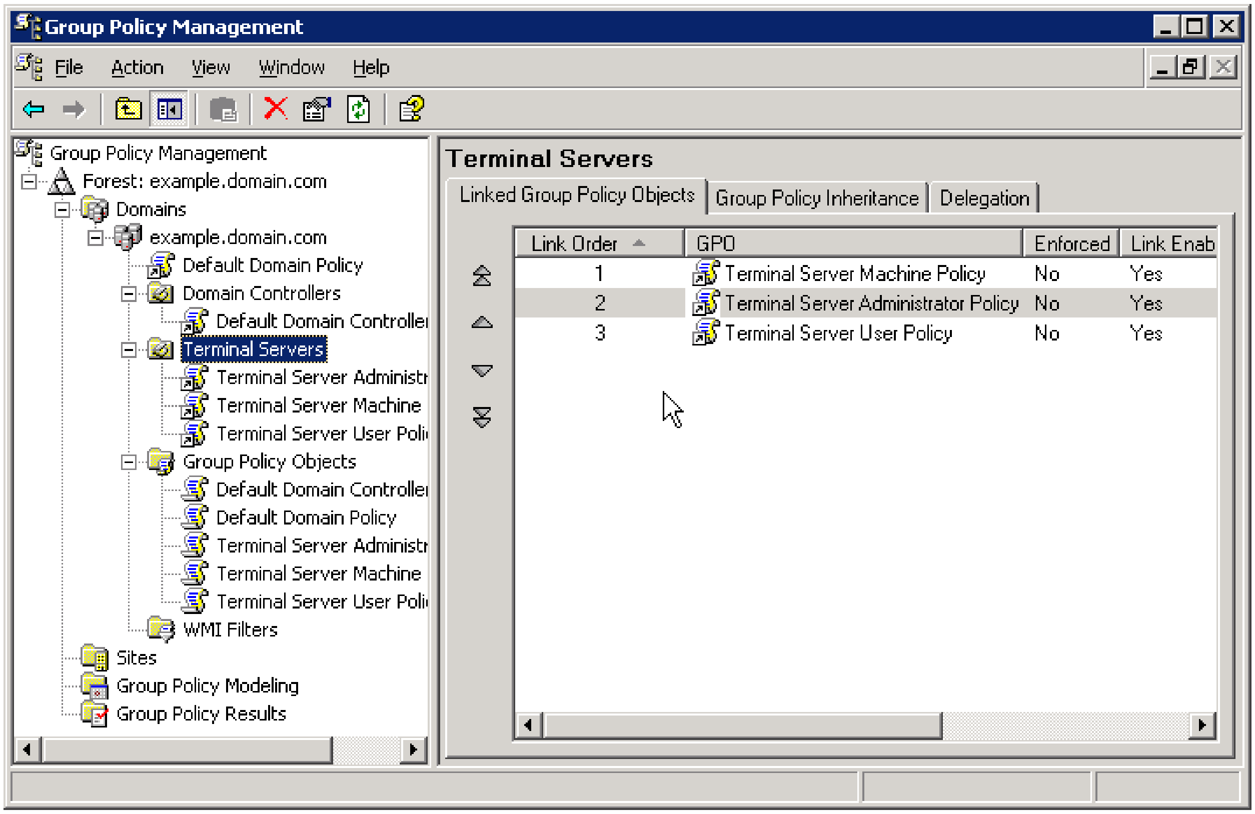
- Folder Redirection: Application Data
- Folder Redirection: Desktop
- Folder Redirection: My Documents
- Folder Redirection: Start Menu
- Remove Map Network Drive and Disconnect Network Drive
- Remove Search button from Windows Explorer
- Disable Windows Explorer's default context menu
- Hide the Manage item on the Windows Explorer context menu

- Hide these specified drives in My Computer (enable this setting for A through D)
- Prevent access to drives from My Computer (enable this setting for A through D)
- Hide Hardware Tab
- Prevent Task Run or End
- Disable New Task Creation
- Disable and remove links to Windows Update
- Remove common program groups from Start Menu
- Disable programs on Settings Menu
- Remove Network and Dial-up Connections from Start Menu
- Remove Search menu from Start Menu
- Remove Help menu from Start Menu
- Remove Run menu from Start Menu
- Add Logoff to Start Menu
- Disable and remove the Shut Down command
- Disable changes to Taskbar and Start Menu Settings
- Hide My Network Places icon on desktop
- Prohibit user from changing My Documents path
- Disable Control Panel
- Disable the command prompt (Set Disable scripts to No)
- Disable registry editing tools
- Disable Task Manager
- Disable Lock Computer

Obviously, if you use all the settings that are listed here, you will have a very restrictive and perhaps unusable environment. For example, Microsoft recommends removing the Map Network Drive command, but if you are in a distributed environment, your users may require this ability to access their documents. Nonetheless, this list provides a good starting point.

I recommend starting with a very restrictive policy, then testing what you can do as you re-enable certain features. Keep in mind that you should use Group Policy to eliminate confusion for users and to help protect the system from undesired and inadvertent activity. You should not rely upon policies alone to secure your server because they leave many loopholes that a malicious user can exploit. For example, if you use the *Prevent access to drives in My Computer* policy as the only method of protecting files on your server, a malicious user could very easily write a harmful batch file. Instead, you should implement restrictive NTFS permissions to protect the file system and use the *Prevent access to drives in My Computer* policy as a way of preventing a user from mistaking the C drive of the server for the C drive of his or her client device. The good news is that if you are running the server in Full Security mode, most key areas of the file system and registry are protected by default.

You also need to be careful to create separate policies for your users and systems administrators. Obviously, an administrator needs certain features that your users do not. To create separate policies, create two GPOs, one for users that enables all of the restrictions you want to implement, and a second for administrators that disables the restrictions on features that you want to allow them to use. Set the delegation on the user policy to apply to Authenticated Users, and set the administrators' policy to apply only to the domain group that contains your administrative accounts. Finally, place the administrators' policy above the user policy in the processing order. Figure 4.10 shows an example of this setup.



- Group Policy Management
  - Forest: example.domain.com
    - Domains
      - example.domain.com
        - Default Domain Policy
        - Domain Controllers
          - Default Domain Controlle
        - Terminal Servers
          - Terminal Server Administr
          - Terminal Server Machine
          - Terminal Server User Poli
        - Group Policy Objects
          - Default Domain Controlle
          - Default Domain Policy
          - Terminal Server Administr
          - Terminal Server Machine
          - Terminal Server User Poli
        - WMI Filters
    - Sites
    - Group Policy Modeling
    - Group Policy Results

Terminal Servers

Linked Group Policy Objects | Group Policy Inheritance | Delegation

Link Order	GPO	Enforced	Link Enab
1	Terminal Server Machine Policy	No	Yes
2	Terminal Server Administrator Policy	No	Yes
3	Terminal Server User Policy	No	Yes

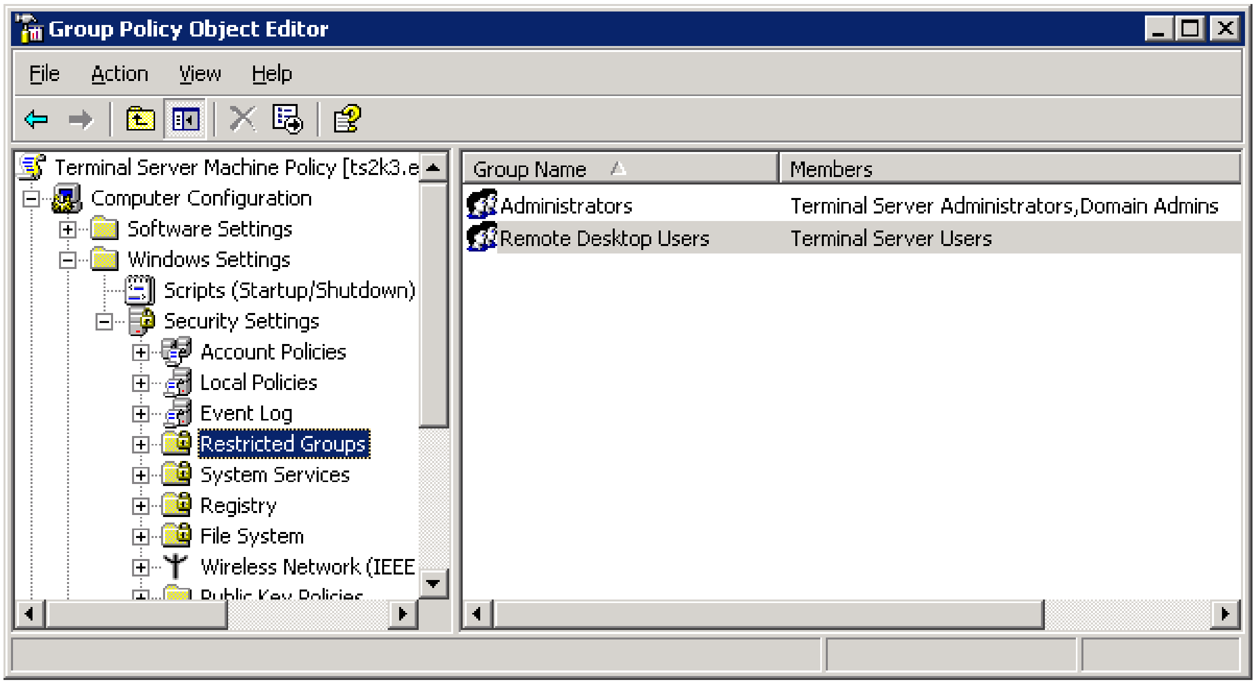
Figure 4.10: Using link order to give administrators a less restrictive policy.

I also recommend keeping your User Configuration in separate policies than your Computer Configuration. Doing so makes it easier to manage changes and keep the same computer settings across multiple user policies. If you choose to do so, be sure to disable the User Configuration settings in the machine policy and vise-versa. Disabling these settings optimizes Group Policy processing by preventing the processing of GPOs that have no relevant settings enabled.

Restricted Groups

By now, you are well aware that membership in the Remote Desktop Users group is critical to using a terminal server. For this reason, I recommend managing this group on all of your terminal servers through Group Policy.

Restricted Groups allows you to manage members of a local machine group through a domain GPO. This way, when a new terminal server is added to the domain, the server immediately inherits the proper domain groups into its Remote Desktop Users group. You can manage the local Administrators group in the same way. Figure 4.11 shows the Restricted Groups node of the Group Policy Object Editor.



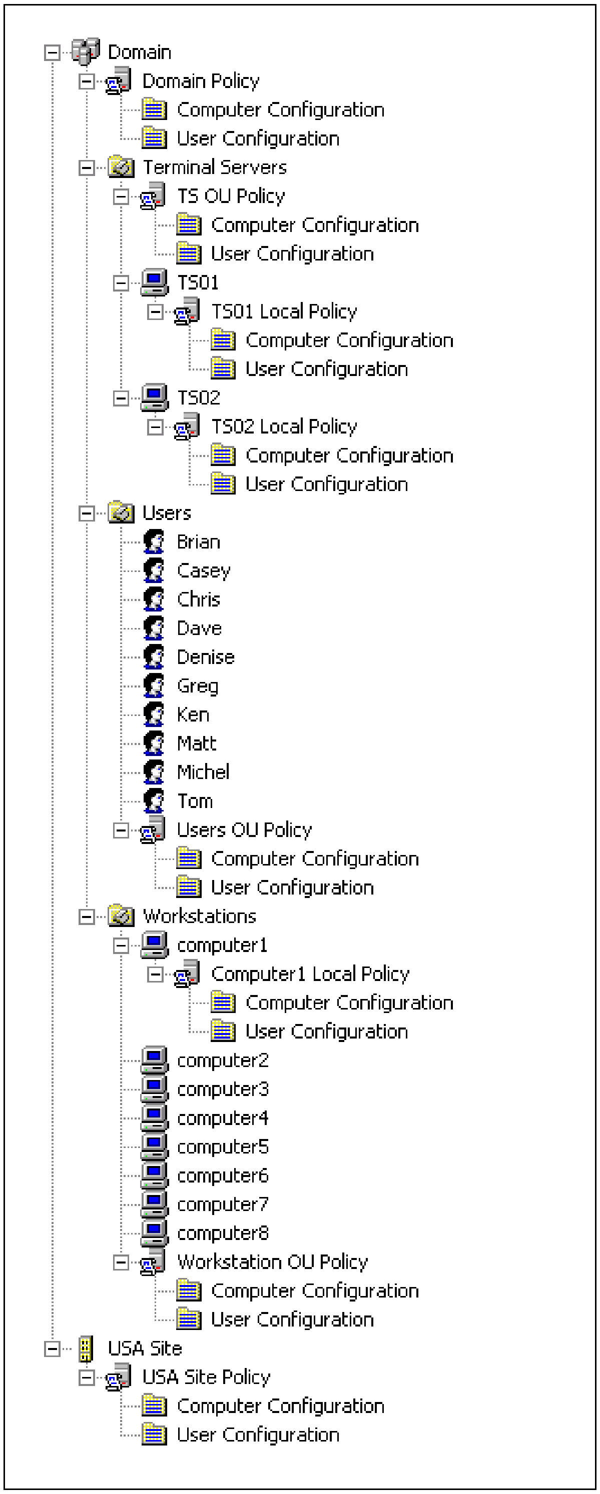
Group Name	Members
Administrators	Terminal Server Administrators,Domain Admins
Remote Desktop Users	Terminal Server Users

Figure 4.11: Managing groups via GPOs.

Standard Group Policy Processing Order

The order that GPOs are applied is very important. Because you can have multiple policies in effect for a given user or computer, the Resultant Set of Policy is what will ultimately determine the settings in effect. To determine the final settings that will be applied to a user, you must look at all the GPOs that are being applied to that user. GPOs are applied in a fixed order: local, site, domain, OU. Machine settings and user settings are processed separately, although they can both come from the same GPOs.

To understand how GPO processing works, let's look at a theoretical domain infrastructure and walk through GPO processing as it happens. Figure 4.12 shows an illustration of an example domain and its GPOs. For simplicity, I have applied only one GPO at each level. In many implementations, you will have multiple GPOs at the site, domain, and OU levels. These GPOs will all have to be processed in order to produce a Resultant Set of Policy.



- Domain
  - Domain Policy
    - Computer Configuration
    - User Configuration
  - Terminal Servers
    - TS OU Policy
      - Computer Configuration
      - User Configuration
    - TS01
      - TS01 Local Policy
        - Computer Configuration
        - User Configuration
    - TS02
      - TS02 Local Policy
        - Computer Configuration
        - User Configuration
  - Users
    - Brian
    - Casey
    - Chris
    - Dave
    - Denise
    - Greg
    - Ken
    - Matt
    - Michel
    - Tom
    - Users OU Policy
      - Computer Configuration
      - User Configuration
  - Workstations
    - computer1
      - Computer1 Local Policy
        - Computer Configuration
        - User Configuration
    - computer2
    - computer3
    - computer4
    - computer5
    - computer6
    - computer7
    - computer8
    - Workstation OU Policy
      - Computer Configuration
      - User Configuration
- USA Site
  - USA Site Policy
    - Computer Configuration
    - User Configuration

Figure 4.12: An example domain and its GPOs.

Let's start by looking at standard GPO processing by booting up computer1. During the boot process, security settings are applied for the computer in the following order:

1. Local—The Computer Configuration settings from the cComputer1 Local Policy are applied.
2. Site—The Computer Configuration settings from the USA Site Policy are applied.
3. Domain—The Computer Configuration settings from the Domain Policy are applied.
4. OU—The Computer Configuration settings from the Workstations OU's Workstation OU Policy are applied. The OU policies are determined by the OU that contains the object in question. In this case, computer1 is in the Workstations OU, so policies linked to that OU will be processed.

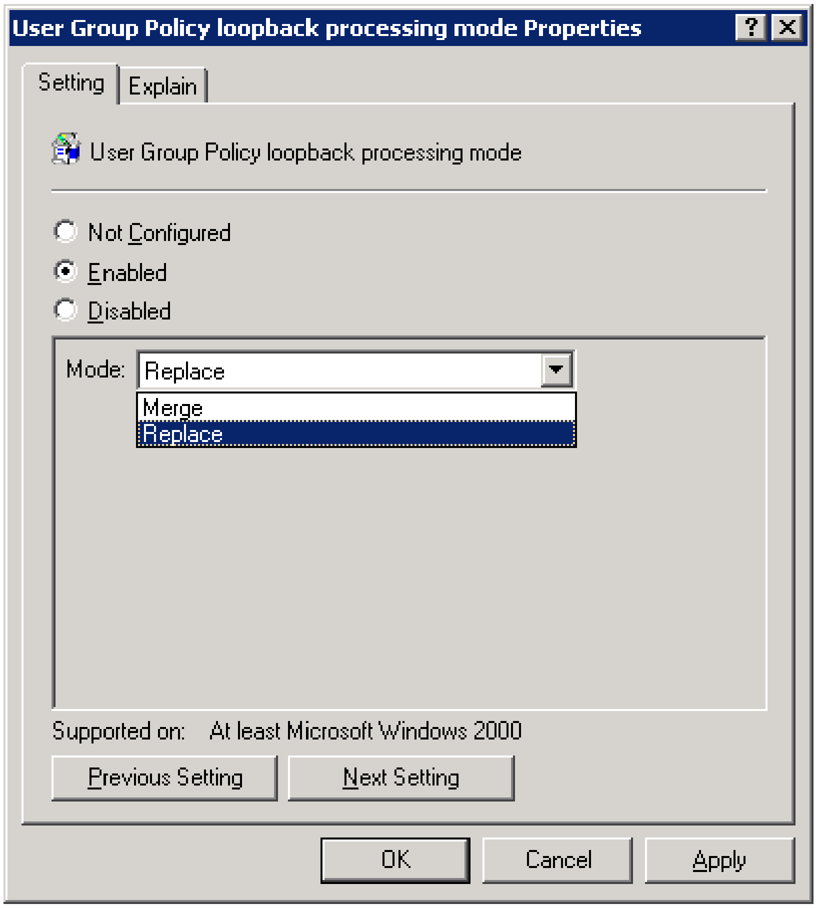
Now that computer1 has a complete configuration, let's have user Greg log on to computer1 so that the User Configuration settings are processed:

1. Local—The User Configuration settings from the Computer1 Local Policy are applied.
2. Site—The User Configuration settings from the USA Site Policy are applied.
3. Domain—The User Configuration settings from the Domain Policy are applied.
4. OU—The User Configuration settings from the Users OU's Users OU Policy are applied. In standard processing mode, Greg will always receive his User Configuration from this policy regardless of which OU contains the computer onto which he is logged on.

In standard processing mode, because there are no user objects in the Workstations OU, the User Configuration of the Workstation OU Policy is never applied. Also, in standard processing mode, Greg will receive the same resultant set of user policy regardless of to which computer he logs on. This functionality is advantageous in a large workstation-based domain where Greg may need to use computers in another department's OU. However, if you want Greg to receive a more restrictive policy when logging on to a specific computer (a terminal server, for example), you will need to implement loopback policy processing.

Loopback Group Policy Processing Order

Loopback processing allows us to take advantage of User Configuration settings from GPOs linked to the OU that contains the computer that is being accessed. As Figure 4.13 shows, there are two modes of loopback processing: Replace and Merge. Merge mode instructs the system to first apply the User Configuration from the Users OU policy (the standard processing order), then apply the User Configuration from the Computers OU policy. The Resultant Set of Policy is the combination of both sets of GPOs. Replace mode instructs the system to ignore GPOs from the Users OU altogether and only apply User Configuration settings from the Computers OU policy.



User Group Policy loopback processing mode Properties

User Group Policy loopback processing mode

○ Not Configured

● Enabled

○ Disabled

Mode: Replace

Supported on: At least Microsoft Windows 2000

Figure 4.13: Group Policy loopback mode options.

In the previously discussed example domain, we can enable loopback policy processing in merge mode on the terminal server TS01. In this scenario, the Computer Configuration is applied as usual, but when Greg connects to TS01, his User Configuration processing is applied in the following order:

1. Local—The User Configuration from the Computer1 Local Policy is applied.
2. Site—The User Configuration from the USA Site Policy is applied.
3. Domain—The User Configuration from the Domain Policy is applied.
4. OU—The User Configuration from the Users OU's Users OU Policy is applied.
5. OU Loopback—The User Configuration from the Terminal Servers OU's TS OU Policy is applied.

In this mode, Greg could receive his Internet Explorer (IE) proxy server settings from the Users OU Policy, but have the Shut Down command removed from his Start menu by the Terminal Servers OU's TS OU Policy. Merge mode has the advantage of being able to place global settings in the Users OU Policy and only apply lockdowns in the Terminal Servers OU's TS OU Policy. The disadvantage is that in this mode, you need to keep track of user settings in two GPOs.

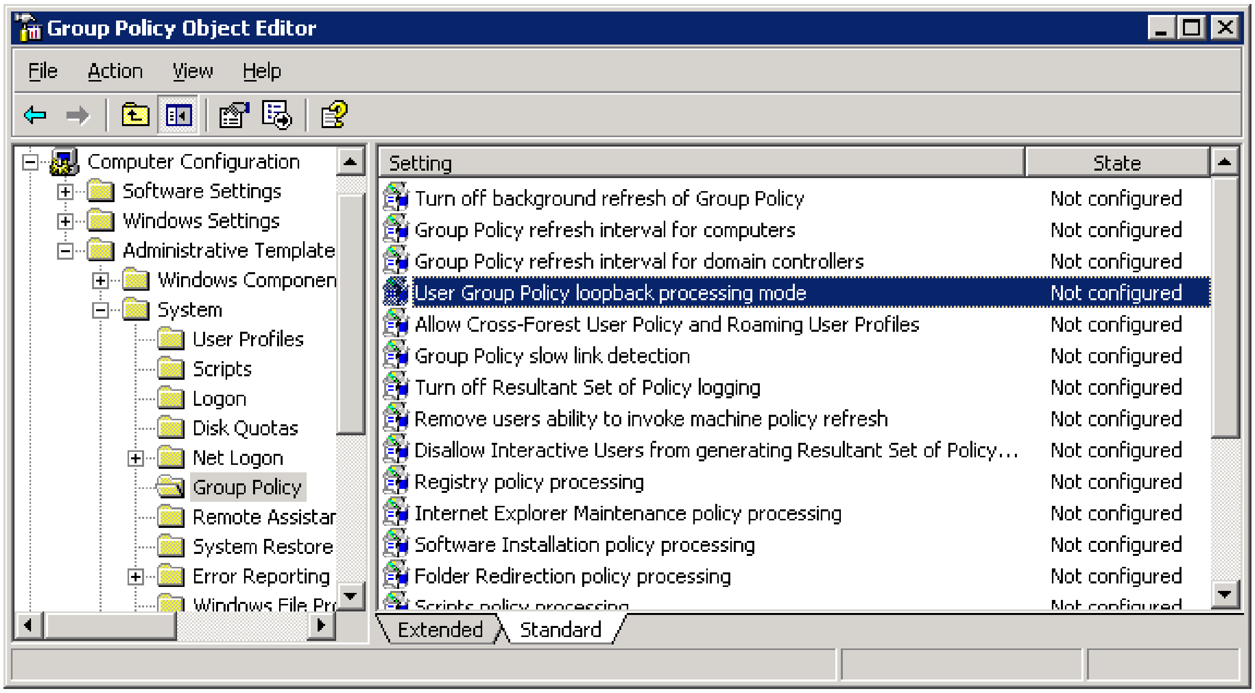
In loopback replace mode, the User Configuration from the Users OU Policy is ignored:

1. Local—The User Configuration from Computer1 Local Policy is applied.
2. Site—The User Configuration from the USA Site Policy is applied.
3. Domain—The User Configuration from the Domain Policy is applied.
4. OU Loopback—The User Configuration from the Terminal Servers OU's TS OU Policy is applied.

This mode simplifies GPO processing by placing all settings into the Terminal Servers OU's TS OU Policy, but it will force you to keep some settings in this policy in sync with changes made to the Users OU Policy. For example, if you are configuring IE's proxy server configuration through Group Policy, the settings applied in GPOs linked to the Users OU need to match those in GPOs linked to the Terminal Servers OU (assuming your proxy settings are the same for workstations and terminal servers). If the proxy server changes, you will need to update both policies.

Enabling Loopback

Loopback processing is enabled through the Computer Configuration section of the Group Policy Object Editor (see Figure 4.14). You can enable loopback processing either in the local machine policy on the terminal server or through any of the GPOs being applied to your terminal servers.



Setting	State
Turn off background refresh of Group Policy	Not configured
Group Policy refresh interval for computers	Not configured
Group Policy refresh interval for domain controllers	Not configured
User Group Policy loopback processing mode	Not configured
Allow Cross-Forest User Policy and Roaming User Profiles	Not configured
Group Policy slow link detection	Not configured
Turn off Resultant Set of Policy logging	Not configured
Remove users ability to invoke machine policy refresh	Not configured
Disallow Interactive Users from generating Resultant Set of Policy…	Not configured
Registry policy processing	Not configured
Internet Explorer Maintenance policy processing	Not configured
Software Installation policy processing	Not configured
Folder Redirection policy processing	Not configured
Scripts policy processing	Not configured

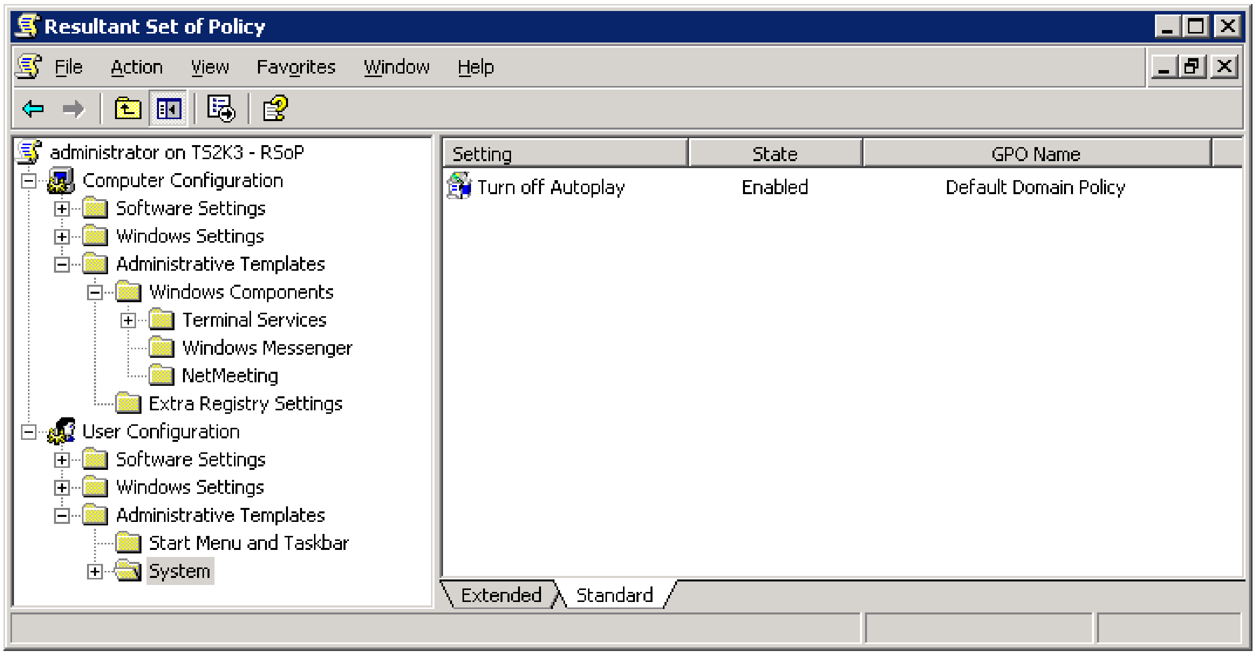
Figure 4.14: The Group Policy Object Editor's loopback setting.

Resultant Set of Policy

With a complex domain that employs many Group Policies and combinations of standard and loopback processing, it becomes very difficult to keep track of the net effect of the GPOs and to fully predict the result of moving a user or computer from one OU to another. To help with this challenge, Microsoft provides the Resultant Set of Policy tool (RSOP.MSC).

You can access this tool from a Run command to retrieve the RSOP of the current user and computer—this information is helpful in identifying the specific GPO that is configuring a questionable setting—or through the GPMC to assist in testing scenarios of users logging on to specific computers. The GPMC also provides access to the Group Policy Modeling tool, which you can use to perform what-if scenarios before making changes to Group Policies or moving objects.

RSOP.MSC not only shows you the final net effect of all policies applied to a user or computer but also lets you access a specific setting and see all the policies that configured it along the way. Figure 4.15 shows the Resultant Set of Policy tool's interface.



Setting	State	GPO Name
Turn off Autoplay	Enabled	Default Domain Policy

Figure 4.15: The Resultant Set of Policy tool.

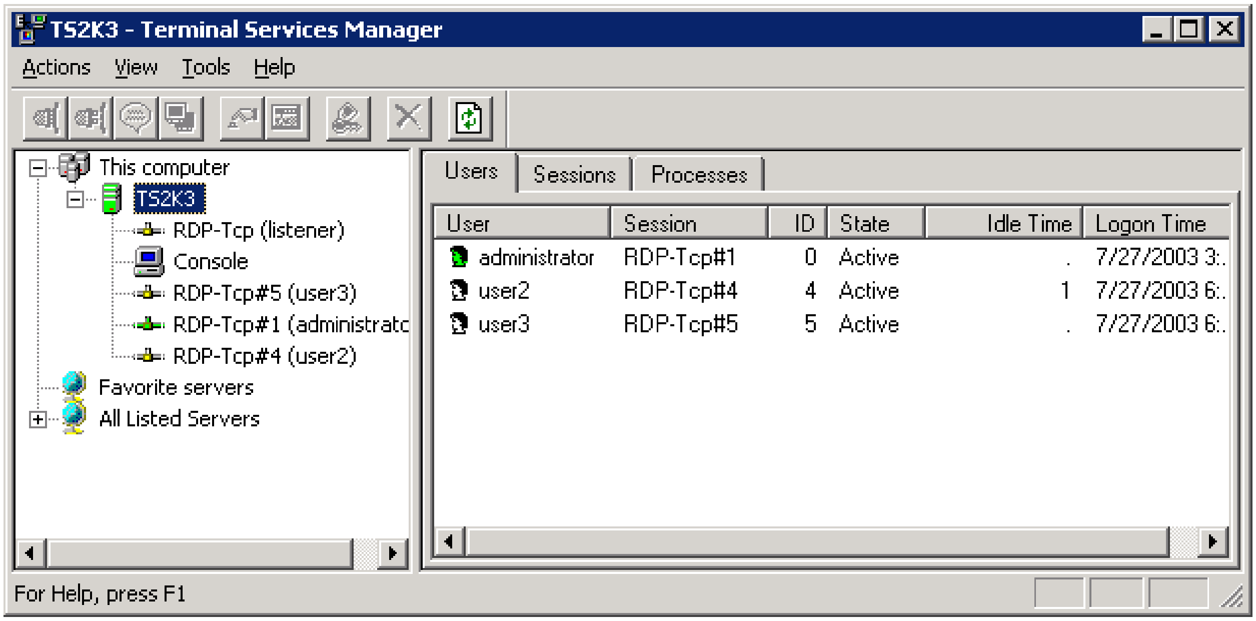
Managing User Sessions

If you're used to a workstation-centric infrastructure, you know that it is a difficult task to remotely diagnose and correct problems for your users. You probably have used such tools as Microsoft Systems Management Server (SMS) remote control and Windows XP Remote Support to work on a user's computer, and you know how to connect to a network registry to modify settings for your users.

In a Terminal Services environment, these tasks are made easier and more straightforward. Rather than tracking down a specific workstation on a remote node of your network, you and your users are both logged on to the same computer. And, because you are both utilizing RDP to transmit KVM data, you can easily tap into the stream to provide assistance. To show you the various support techniques, I will first introduce you to the support utilities.

Terminal Services Manager

When you launch the Terminal Services Manger administrative tool, you are presented with a list of all servers that have Terminal Services enabled in the domain. Using this tool, you can easily see to which servers users are connected, from which client devices they're accessing the servers, and which processes and applications they are running in their sessions. Figure 4.16 shows the Terminal Services Manger interface.



User	Session	ID	State	Idle Time	Logon Time
administrator	RDP-Tcp#1	0	Active	.	7/27/2003 3:.
user2	RDP-Tcp#4	4	Active	1	7/27/2003 6:.
user3	RDP-Tcp#5	5	Active	.	7/27/2003 6:.

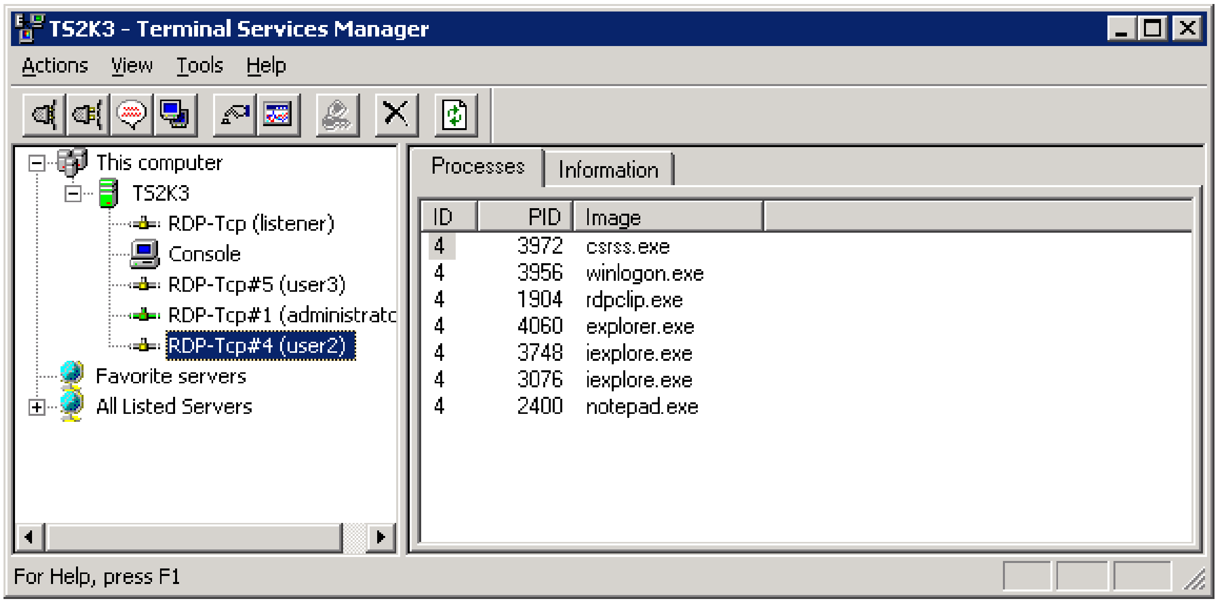
Figure 4.16: The Terminal Services Manager interface.

If you are familiar with the Win2K Terminal Services Manager, you will see some nice improvements in the version included with WS2K3. First, the new version offers a *This computer* node, which gives you quick access to sessions on the server to which you are logged on. Second, there is a *Favorite Servers* node, which allows you to quickly access specific terminal servers that you frequently administer. Finally, for performance, the *All Listed Servers* node is not expanded by default, so you no longer have to wait for all terminal servers to be enumerated before using the tool.

Once you highlight a specific server, the tool shows you a list of all user sessions on that terminal server. It also shows you each session's status using the following categories:

- Active—The user is actively sending keyboard or mouse information to the server.
- Idle—The user has not moved the mouse or entered a keystroke in a given period of time.
- Disconnected—The user has disconnected from the server, but has left the session running for later reconnection.

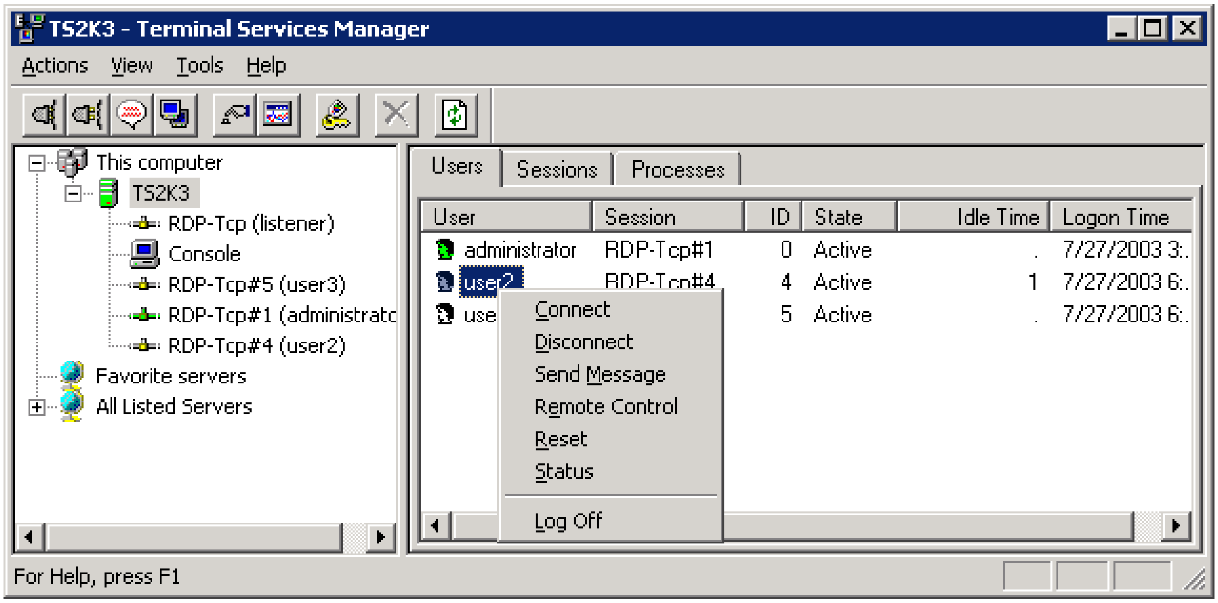
If you highlight a user's session in the left pane, the results pane will show you all processes that user has running on the server. In this pane, you can end a hung process for the user (see Figure 4.17).



ID	PID	Image
4	3972	csrss.exe
4	3956	winlogon.exe
4	1904	rdpclip.exe
4	4060	explorer.exe
4	3748	iexplore.exe
4	3076	iexplore.exe
4	2400	notepad.exe

Figure 4.17: Active processes in a user's session.

The Information tab of the right pane will present you with the user's client device name and IP address as well as the version number of the RDP client that he or she is running, the screen resolution, and the encryption level. This information can assist you in troubleshooting connectivity problems. If you right-click a user's session, you are presented with the seven options that Figure 4.18 shows.



User	Session	ID	State	Idle Time	Logon Time
administrator	RDP-Tcp#1	0	Active	.	7/27/2003 3:..
user2	RDP-Tcp#4	4	Active	1	7/27/2003 6:..
use		5	Active	.	7/27/2003 6:..

Figure 4.18: Options available to administrators from the Terminal Services Manager interface.

These commands allow you to interact with the user's session in several ways:

- Connect—Lets you connect to another session that you have established on a server.
- Disconnect—Disconnects the user from a session, but leaves the session running on the server.
- Send Message—Sends a pop-up message to the user.
- Remote Control—Allows you to view or interact with the user's session without disconnecting the user. The user sees any activity that you perform while remote controlling, and you can observe the user's activity as well.
- Reset—Kills the session.
- Status—Shows a status window of network activity between the server and client device.
- Log Off—Forces the user to log off of the session.

> The Log Off option will perform a graceful logout and upload the user's profile to the central profile directory. However, it will not give the user any opportunity to save work in progress.

Remote Control

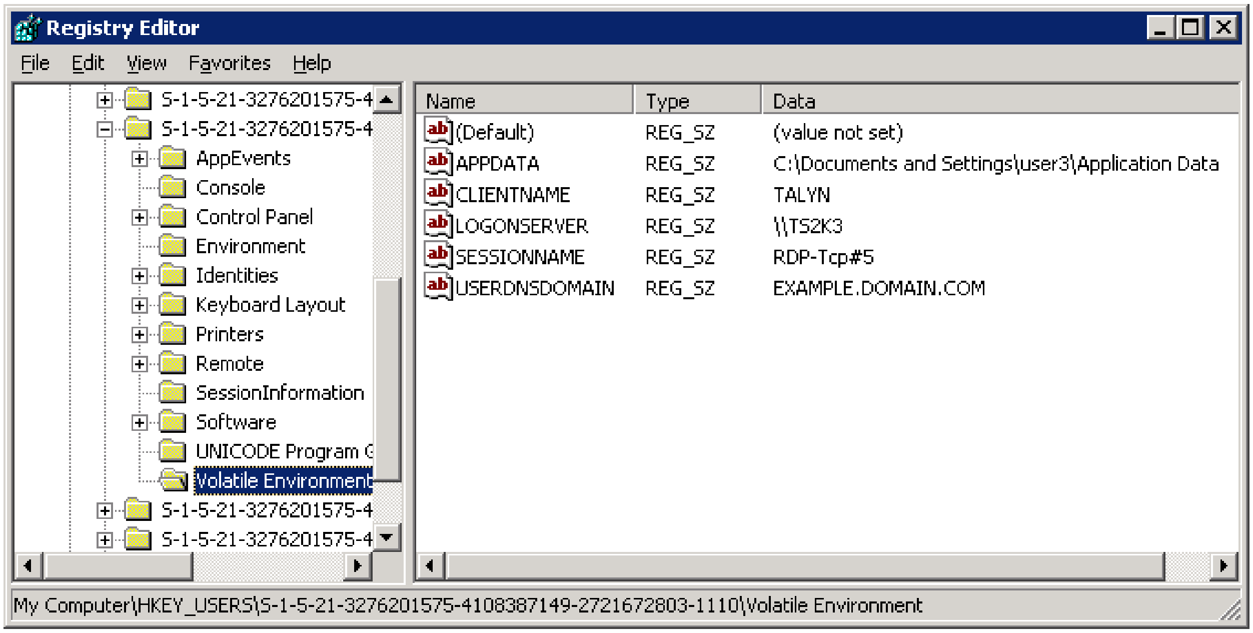
When you select Remote Control from the Terminal Services Manager interface, you temporarily disconnect from your session and are connected to the user's session. RDP now sends all video information to both your machine and the user's client device and receives keyboard and mouse movements from both of you (if you have configured remote control with the interact privilege).

While remote controlling, the user can observe you while you launch applications, change settings, and so on, and you can observe the user's activity. One thing to remember is that any restrictions that you have placed on the user through Group Policy are in effect while remote controlling, so if you have disabled registry editing in the user's User Settings GPO, you will not be able to launch REGEDIT while shadowing the user's session.

Registry Editing

There are some support issues that may require you to edit a user's registry. In a workstation environment, this feature requires you to connect to the remote registry on the user's workstation. On a terminal server, you are sharing the same registry with your users. There is only one HKEY_LOCAL_MACHINE subkey for all users, and each session's HKEY_CURRENT_USER subkey can be seen under HKEY_USERS.

Each user's registry is listed by SID. The quickest way to find a specific user, assuming you don't know the user's SID, is to look in the Volatile Environment subkey for each user. This subkey contains the APPDATA variable, which will list the username in its path, as Figure 4.19 shows. Any changes you make here are immediately seen by the user.



Name	Type	Data
(Default)	REG_SZ	(value not set)
APPDATA	REG_SZ	C:\Documents and Settings\user3\Application Data
CLIENTNAME	REG_SZ	TALYN
LOGONSERVER	REG_SZ	\\TS2K3
SESSIONNAME	REG_SZ	RDP-Tcp#5
USERDNSDOMAIN	REG_SZ	EXAMPLE.DOMAIN.COM

My Computer\HKEY_USERS\S-1-5-21-3276201575-4108387149-2721672803-1110\Volatile Environment

Figure 4.19: The HKEY_USERS subkey showing user3's HKEY_CURRENT_USER registry subkey.

Command-Line Utilities

There are several command-line utilities that you can use to help manage your servers. Most have direct counterparts in Terminal Services Manager, but the command-line versions can be useful if you are writing administrative scripts. The following list highlights command-line utilities:

- CHANGE USER {/Install /Execute}—The CHANGE USER command is used to switch the server between install and execute mode for application installation.
- CHANGE LOGON {/Enable /Disable}—The CHANGE LOGON command can disable any new logons from being accepted by the server.
- Query {Process | Session | Termserver | User}—The Query command presents similar information as the Terminal Services Manager tool presents; this command lists active processes by session, active sessions, available terminal servers, and current users.
- TSSHUTDN—The TSSHUTDN command is used to shut down or reboot a terminal server. This command, unlike the Shut Down command from the Start menu, gives your users a warning that the server is being shut down so that they can save their work. It then forces a logoff for each session, and finally shuts down the server.

> You can shorthand the Query User command to QUSER to get a quick list of active sessions on a terminal server.

> There are several third-party tools that can help you manage and lock down your terminal servers, plug the loopholes that GPO-based lockdowns leave behind, and manage user profiles and printing. For example, triCerat Simplify Profiles lets you save and restore, set, and delete end-user registry settings.

Summary

In this chapter, we explored administrative aspects of Terminal Services, including the multiple options available for configuring user session parameters and timeouts. In addition, I introduced you to terminal server profiles and explained when and how to use them. We dove into AD-based administration, taking you through the often complex topic of Group Policy processing and how to use loopback processing with terminal servers. Finally, I covered the tools used to manage and support user sessions on terminal servers.

In Chapter 5, I will dive into the process of installing applications on your terminal servers, including the user logon process and how it takes advantage of ROOTDRIVE and application compatibility scripts. We'll look at how to manage application life cycles and how you can use IntelliMirror to centrally deploy applications to your server farm.