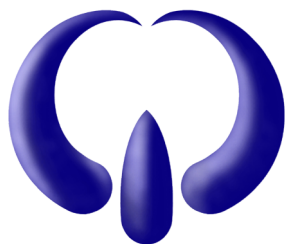




realtimepublishers.comtm

The Definitive Guidetm To

Windows Server 2003 Terminal Services



triCerat
Software®

Gresyon Mitchem

Chapter 2: Installing and Configuring the Terminal Server Role.....	23
Terminal Services Deployment Scenarios	23
Desktop Replacement	24
Remote Access.....	25
ASP	26
Installing the Terminal Server Role.....	26
Configuring the Terminal Server Role	30
Terminal Services Configuration Administrative Tool	30
Permission Compatibility.....	31
Licensing.....	32
Restrict Each User to One Setting	32
Group Policy–Based Configuration.....	37
Additional Configuration Settings	39
Installing and Configuring the Remote Desktop Connection Client	41
Remote Desktop Connection Client.....	41
Remote Desktop Web Connection.....	43
Remote Desktops Administrative Tool.....	44
Summary	45

Copyright Statement

© 2003 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

Chapter 2: Installing and Configuring the Terminal Server Role

This chapter will take you through the steps of adding the terminal server role to WS2K3. I'll introduce you to the settings used to configure a terminal server via the administrative tools, Group Policy, and registry editor, and even give you a few system tweaks to help improve your server's performance. Finally, I will give you an in-depth look at the Remote Desktop Connection client, and the new version of the Terminal Server Advanced Client (TSAC)—Remote Desktop Web Connection. I'll begin by exploring the most common reasons for deploying Terminal Services.

Terminal Services Deployment Scenarios

What are the biggest challenges you face managing a Windows desktop environment? Answer truthfully, and I'm sure these are among your top five:

- Software deployment
- Virus protection
- Software updates

By using Terminal Services, you can greatly reduce the difficulty of these tasks. WS2K3's terminal server role can enable you to centralize software, decrease the number of Windows systems in the environment, and prevent exposure to viruses by centrally managing virus scanner updates and creating a single point of entry for remote access users.


There are three basic models for utilizing Terminal Services:

- Desktop replacement—Remove the Windows PC from the user's desk, and replace the PC with a thin-client device.
- Remote access—Provide users in a remote location access to either a complete desktop environment or individual applications over either a WAN link or Remote Access Service (RAS) connection.
- Application service provider (ASP)—Provide access to individual applications to users at their regular workstation without installing the applications locally on users' PCs.


Desktop Replacement

At its most pervasive implementation, Terminal Services can allow an IT department to completely eliminate PCs from users' desks. This model provides many benefits, including elimination of end-node support, rapid deployment of new or upgraded software, reduction in power consumption, and added security. Depending on your corporate IT architecture, desktop replacement can also reduce bandwidth requirements and eliminate the need for servers in remote offices:

- **Elimination of end-node support**—Without PCs on users' desks, there is no longer any need to visit the workstation to configure the OS, install or repair software, assist a user in configuring applications, or replace a defective hard drive. A thin-client's OS is burned into ROM, and applications are installed on the terminal servers. Help desk personnel can provide user assistance via remote control of the terminal server session, and users can replace a damaged device by simply plugging in a new one.

 Most thin-client devices support auto-configuration via Dynamic Host Configuration Protocol (DHCP) and FTP. You add a URL to a DHCP extension, and when the client boots, the client downloads its configuration from the FTP URL. This setup enables even the least tech-savvy users to set up or replace thin clients.

- **Rapid deployment of new or upgraded software**—If you work in a large IT environment, you know how difficult and time consuming deploying software to your users can be. By using Terminal Services and thin clients, you simply install the new software on your servers, and overnight thousands of users will have access to it. There are even third-party utilities that will assist you in deploying software to all of your terminal servers at once.
- **Reduction in power consumption**—Thin-client devices have no moving parts and are completely solid-state, so they typically consume about 10 percent of the power that a normal Wintel PC consumes. With rising electricity costs, this reduced consumption can provide a major cost-savings to your company.
- **Added security**—If a standard PC is stolen, you risk losing important and sensitive data stored on its local hard disk, and you must pay to replace the computer. With the desktop replacement model, there is no data stored on the end-node device, and the cost of replacement is about half that of a normal PC.

 There are many players in the thin-client device market (for example, Wyse Technologies' Winterm and Neoware EON). These devices can use any embedded OS at their core (Windows CE, embedded Linux, and so on).


There are, however, some potential drawbacks to the desktop replacement model, including limited adaptability to one-off applications, reduction in user settings personalization, and increased initial deployment costs:

- Limited adaptability to one-off applications—If you have a very small group of users who need an application, with the desktop replacement model, you'll no longer have the freedom to install the application on only those users' desktops. You'll be forced to integrate the application into your Terminal Services infrastructure. Because of this limitation, the desktop replacement model is best suited to homogeneous computing environments.
- Reduction in user settings personalization—If your users are accustomed to personalizing their workstations with wallpaper and screen savers or have the ability to install their own software, you'll have a small battle on your hands when they're restricted from performing some of these customizations.
- Increased initial deployment costs—If you already have a large user population and each user has a PC, the initial setup costs of purchasing the thin-client devices and the robust servers needed for terminal servers can seem a little overwhelming. However, in the long term, the reduction in TCO will more than make up for the initial investment. Opening a new office or call center is a perfect opportunity to implement the desktop replacement model.

Remote Access

If these drawbacks or your corporate culture eliminate desktop replacement as an option, the remote-access model may be a great alternative for you. Most big companies have a large population of remote or nomadic users—telecommuters, executives traveling to satellite offices, and so on. Although laptops provide the ability to work remotely, they don't address the needs of limited-bandwidth connections or remote support. In addition, laptops can be nearly double the cost of a desktop, so your finance department may balk at the thought of providing a laptop for users who only occasionally need remote access to applications.

The remote-access model can provide these users with the ability to access individual applications or even a complete corporate desktop from the Internet (by using the Remote Desktop Web Connection, a Web-based version of the Remote Desktop Connection client) or from their home computers. In addition, the reduced bandwidth requirements of RDP provide improved performance when compared with running laptop-based applications over a slow link.

 Using a terminal server as a portal to the corporate LAN can shield your network from any viruses on the remote computer.


As with any remote-access strategy, you must make security the top priority when considering the remote-access model. Be sure to take the time to educate your network design engineers in the specific needs of the terminal server protocols. In addition, implement a strategy to prevent the abuse of any changes you implement to accommodate the terminal server network traffic.

ASP

When looking at a large-scale deployment of a vertical application, there are many factors to consider:

- Deployment method (Sneakernet, Systems Management Server—SMS, IntelliMirror)
- Workstation system requirements (RAM, disk space, processing power)
- Support and back-out plan in case an installation goes awry
- Bandwidth requirements for client/server or database applications

If the application you are deploying doesn't have complex OLE integration with other applications on the users' desktops, the ASP model might be right for you. Terminal servers, especially when implemented with TSAC or a third-party application publishing product, can give you the ability to provide users the applications they need quickly and easily without touching their workstations. In this model, the application is installed on terminal servers, and users launch it via a client application on their desktops or by using a Web browser.

 I'll go into the details of the ASP model in Chapter 5.

Installing the Terminal Server Role

When an administrator logs on to WS2K3, the Manage Your Server Wizard, which Figure 2.1 shows, provides easy access to the tools needed to install, configure, and manage server roles. Here is where we will begin the process of installing the terminal server role.

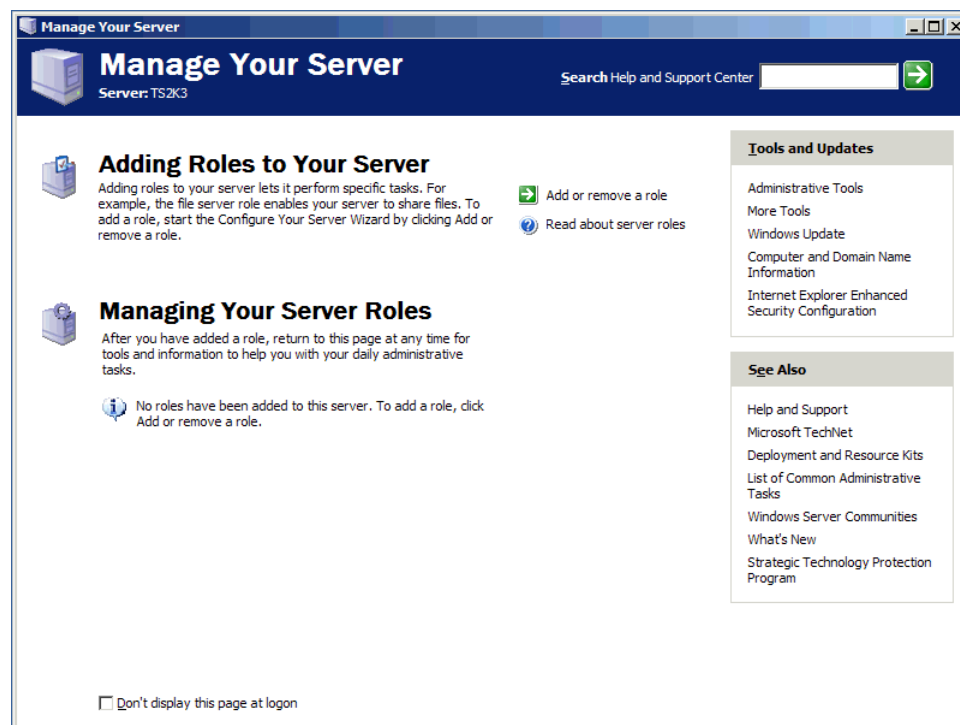


Figure 2.1: The Manage Your Server wizard.

To start, click the *Add or remove a role* link to invoke the Configure Your Server wizard, which Figure 2.2 shows. This wizard outlines the preliminary steps to adding a role. Confirm that you are prepared, and click Next.

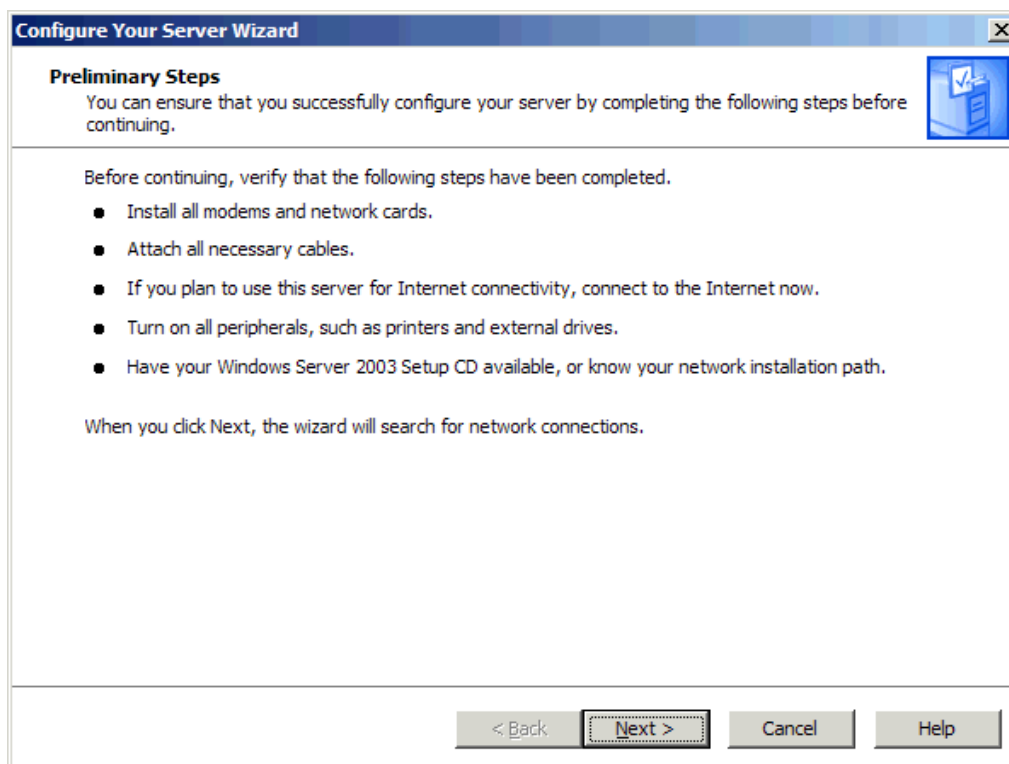


Figure 2.2: Preliminary steps to installing a role.

The wizard then scans your network connections to determine which roles are compatible, then lists all available roles for your server, as Figure 2.3 shows.



Figure 2.3: Detecting your network settings to determine compatible roles.

In the window that results from the scan (see Figure 2.4), you will select the terminal server role, then click Next.

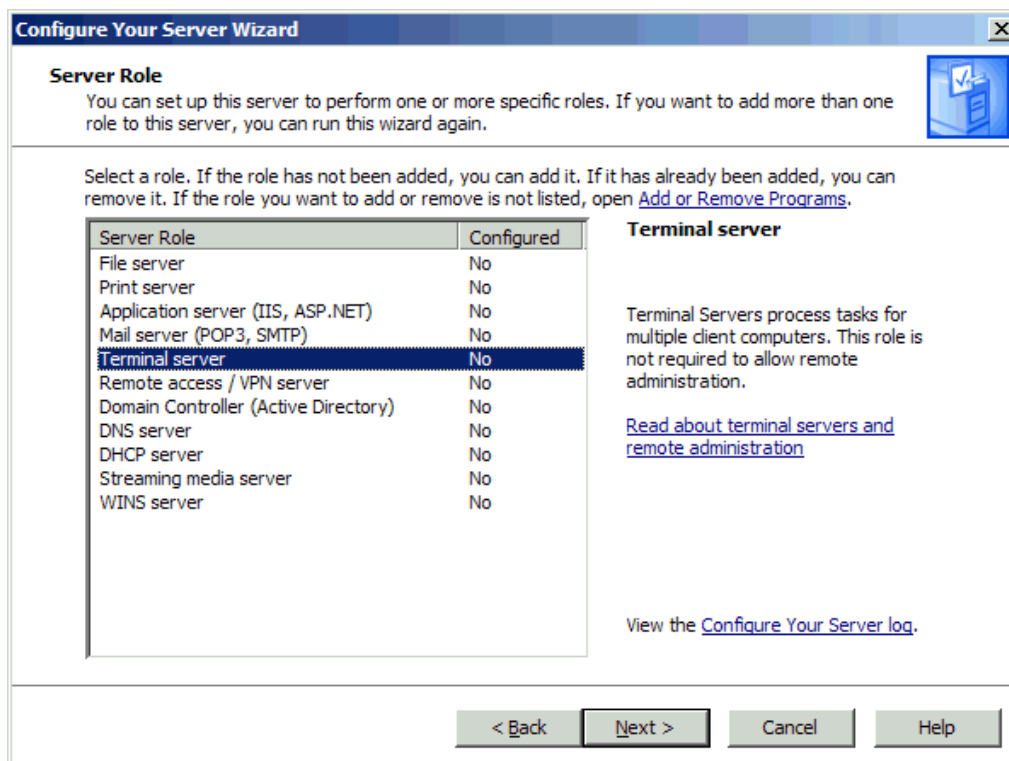


Figure 2.4: The Configure Your Server wizard.

After warning you that adding this role will automatically reboot your server, the wizard will call the Add/Remove Windows Components Control Panel applet and add the required services. When complete, the system will reboot.



There is no option to postpone the reboot when adding a role via the Manage Your Server wizard.

When you log on to the system after the reboot, two windows will be automatically displayed. One window confirms that the terminal server role has been successfully added, as Figure 2.5 illustrates.



Figure 2.5: A successful installation of the terminal server role.

The second window provides a helpful checklist of the common next steps needed to complete the configuration of your terminal server (see Figure 2.6).

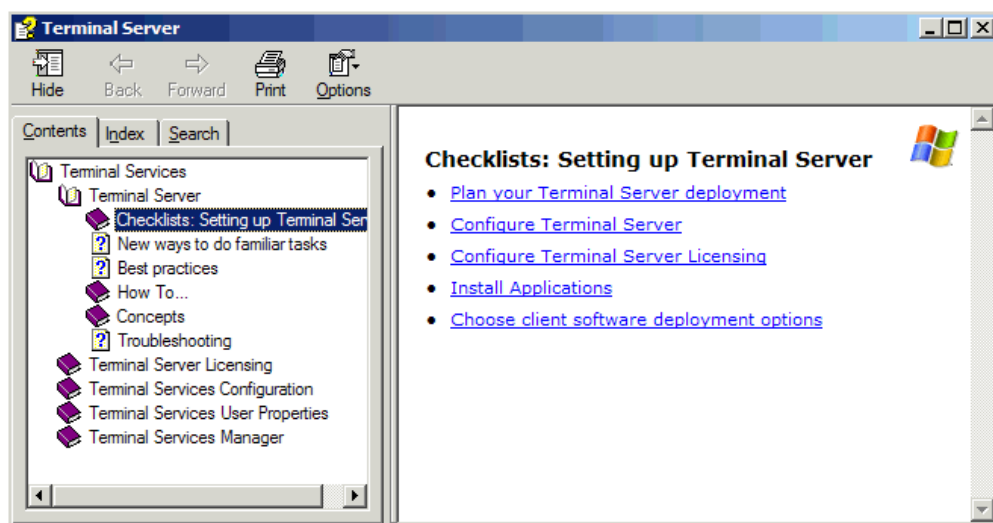



Figure 2.6: The common next steps required for configuring a terminal server.

Configuring the Terminal Server Role

As you can see in Figure 2.6, there are several steps you must take after installing Terminal Services. We explore terminal server licensing in Chapter 1; in this section, I'll discuss how to configure a terminal server.

 The reference materials available in the Plan your Terminal Server Deployment section of the checklist are very helpful, so read through them.

There are two main tools used to configure a terminal server: the Terminal Services Configuration tool and the Group Policy editor.

Terminal Services Configuration Administrative Tool

The main tool used to configure a terminal server is the Terminal Services Configuration administrative tool. With this tool, you can set the permission mode for the server, configure performance options, and configure RDP. You can launch Terminal Services Configuration in one of three ways:

- From the Start menu, under Administrative Tools
- Directly from the Configure Terminal Server wizard checklist
- From the Manage Your Server wizard

Under the Server Settings node, which Figure 2.7 shows, you will find six options.

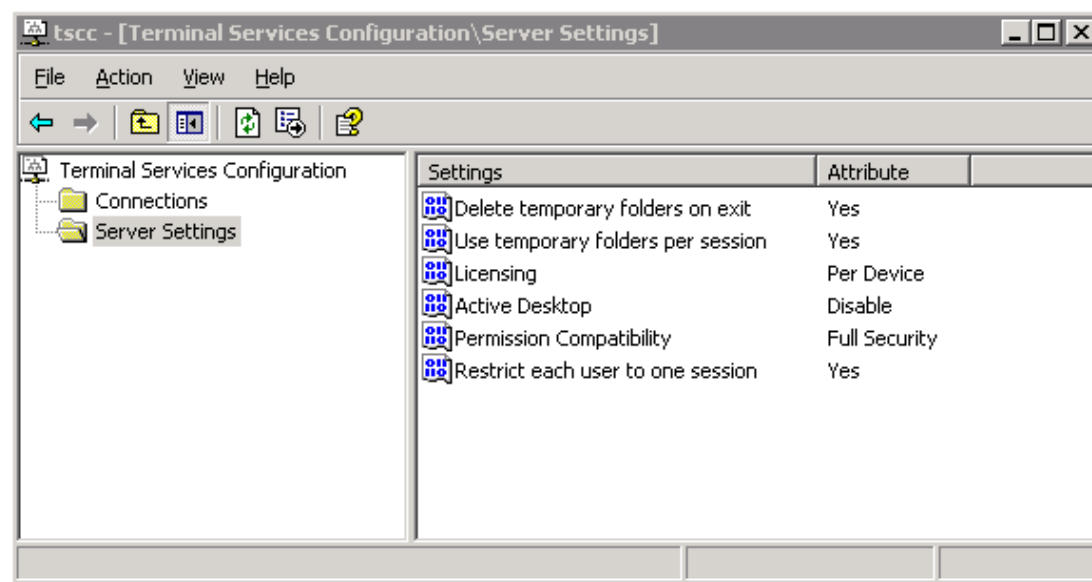


Figure 2.7: The server settings node of the Terminal Services Configuration tool.

Three of these choices—*Delete temporary folders on exit*, *Use temporary folders per session*, and *Active Desktop*—you will most likely leave in the default settings. The following list provides an explanation of these settings:

- **Delete temporary folders on exit**—Each user on a terminal server is given a temp directory. This folder is found in C:\Documents and Settings\\local settings\temp. If this setting is enabled, the temp folder is purged when the user logs off the server.

If you use roaming profiles for your users, and you enable the *Delete cached copies of roaming profiles* Group Policy setting (a common practice on terminal servers), the *Delete temporary folders on exit* setting becomes irrelevant as the entire profile directory is deleted at logoff. However, it is a good idea to leave this setting enabled unless you have an application that requires that temp files are persistent across sessions, in which case, you will also need to disable the Group Policy setting as well.

- **Use temporary folders per session**—With this setting enabled, a new directory is created under the users temp folder for each session the user has on the server. These folders are named with a single digit (\temp\0, \temp\1, and so on). It is a good idea to leave this setting enabled to prevent multiple sessions from interacting.
- **Active Desktop**—Starting with Windows 98, we have had the ability to embed active content (Web pages, animations, news tickers, and so on) on the Windows desktop. To reduce the number of screen redraws being sent to the client from the terminal server, this setting is disabled by default.

The remaining three settings—*Licensing*, *Permission Compatibility*, and *Restrict each user to one setting*—require a little more consideration and understanding. These settings are dependent on your environment and the applications you intend to install on the terminal server. Let's begin by looking at permission compatibility.

Permission Compatibility

In Win2K, you were prompted to select a compatibility mode when installing Terminal Services. The options were *Permissions compatible with Windows 2000 Users* and *Permissions compatible with Terminal Server 4.0 users*. In line with Microsoft's new focus on security, WS2K3 defaults to Full Security mode. This mode is similar to the Win2K Users mode. Under WS2K3 Full Security mode, non-administrators cannot modify the HKEY_LOCAL_MACHINE registry key nor write files to anywhere on the server's hard drive other than their profile directory.

If you encounter applications that will not run under Full Security mode, you may need to change to Relaxed Security mode. Use this option as a last resort, as it opens your server up to inadvertent changes by non-administrators.

 In Chapter 5, I will go over some alternatives to Relaxed Security mode that you can use to enable some older applications to run on Terminal Services.

Licensing

The next setting to address is the licensing mode. This setting controls the type of licenses that the terminal server will request from the license server on behalf of the clients. In most cases, the default setting is Per Device, which means that you will need to install WS2K3 Terminal Server Per Device tokens on your license server. However, if you are upgrading a Win2K terminal server that has Internet Connector Licensing enabled, you'll configure this setting to Per User licensing.

The mode you select is dependent on your environment. If your environment is one in which each user has multiple devices from which they will connect, Per User licensing may be easier to manage and may even save you some money; whereas, if your users share computers, Per Device licensing may be a better option. Perhaps you manage a call center in which one computer is shared by three users, one in each shift. Per Device licensing would mean you would only need one token to cover three users. If you set the server to Per User licensing, it will also validate and accept connections from devices that have already been issued a Per Device token.

Restrict Each User to One Setting

The last setting to be considered is *Restrict each user to one session*. Enabling this setting will prevent users from establishing multiple sessions on the server, which will help conserve resources on the server by only allowing each user to take up the overhead of a single session and run all required applications within that session. Keep in mind that if you are going to be offering direct access to individual applications outside of a desktop environment, your users might need the ability to run more than one application at the same time.


 Citrix MetaFrame supports session sharing. This functionality lets a user launch multiple published applications on the same server without establishing a separate session for each one.

Figure 2.8 shows the connections node of the Terminal Services Configuration administrative tool. Through this node, you configure timeouts, security, and client resource redirection.

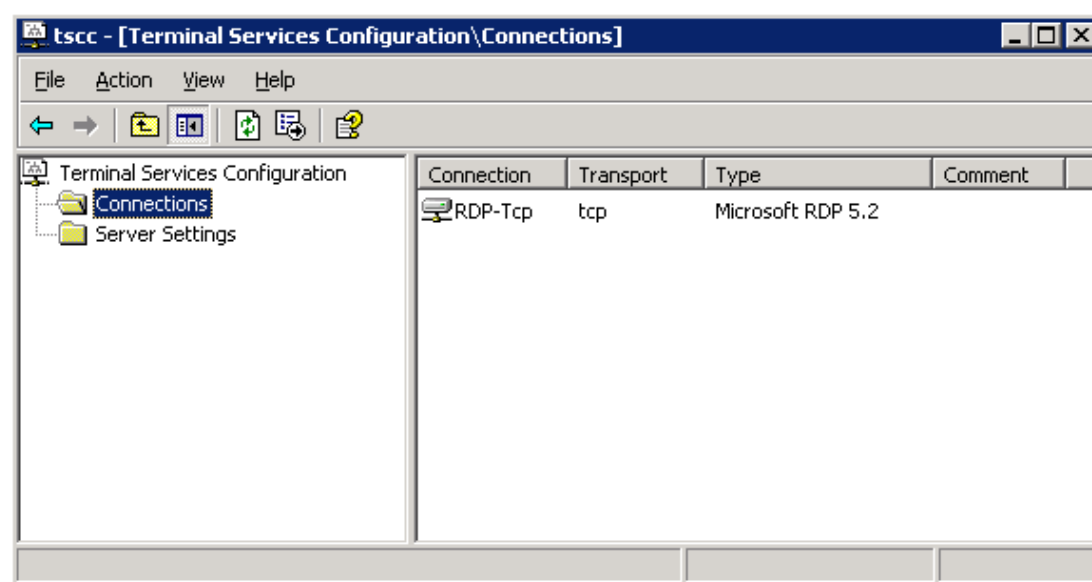


Figure 2.8: The connections node of the Terminal Services Configuration tool.

By default, you will see only one RDP-Tcp connection. If you are using a multi-homed server, you can modify the default connection definition to apply only to one network interface, then create a separate connection definition for your other interfaces. Also, if you have installed Citrix MetaFrame, you will see one or more ICA connections here; it is advisable to use the Citrix Connection Configuration tool to modify settings for the ICA protocol.

By right-clicking the connection, you can disable it entirely, rename it, or access its properties. If you are familiar with the Win2K Terminal Services Configuration tool, the WS2K3 tool's interface looks quite familiar, with the addition of the new features of RDP 5.2 and the new "secure by default" model.

The General tab of the RDP-Tcp properties (see Figure 2.9) lets you add a comment to the connection and to set the encryption level. WS2K3 offers new encryption options:

- Low—All data sent from the client to the server is protected by 56-bit encryption.
- Client Compatible (the default setting)—All data sent between the client and the server is protected by encryption based on the maximum key strength supported by the client.
- High—All data sent between the client and the server is protected by encryption based on the server's maximum key strength. Clients that do not support this level of encryption cannot connect.
- FIPS Compliant—All data sent between the client and the server is protected by using Federal Information Processing Standard (FIPS) 140-1 validated encryption methods.

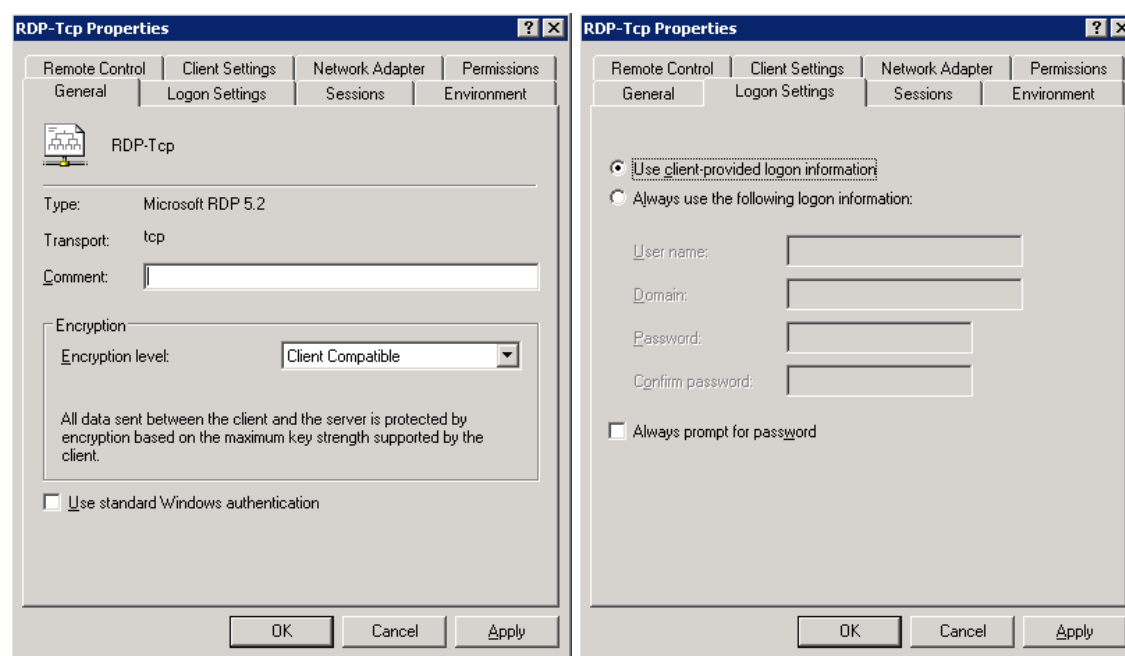


Figure 2.9: The General and Logon Settings tabs of the RDP-Tcp connection properties.

On the Logon Settings tab, which Figure 2.9 also shows, you can control whether to allow users to log on as themselves or specify a single account to automatically log on as users when they connect to the server over RDP. On this tab, you can also select the *Always prompt for password* option, which prompts the user for a password even if one is cached in the Remote Desktop client.



Be careful about setting credentials for automatic logon, as doing so will prevent you from logging on with an administrative account.

Figure 2.10 shows the Sessions and Environment tabs of the RDP connection. In these windows, you set timeouts and reconnection settings as well as an initial program to launch. By default, the settings on both of these tabs are inherited from the parameters set on the user account connecting to the server. If you want to override the user account settings, do so here.

The Sessions tab contains timeouts for disconnected, idle, and active sessions. A disconnected session is one in which the user actively disconnects from the server by either closing the connection window without logging off or selecting Disconnect from the Start menu. An idle session is one in which the user has left the connection window open, but has not executed any mouse clicks or keystrokes in a given period of time. When a session loses its network connection or reaches the idle timeout, you can specify whether to immediately end the session or to treat the session as disconnected.

The Environment tab lets you specify a specific program to launch when a client connects to the server. You must specify both the path and executable name. If you configure this setting, when any user, including an administrator, connects to the server, the specified program will run instead of a Windows Explorer desktop. Many administrators have made the mistake of thinking that this setting is like the Startup folder in the Start menu, automatically launching a program when the user logs onto the desktop. Such is not the case—configuring this setting *replaces* the Explorer shell with the program specified.

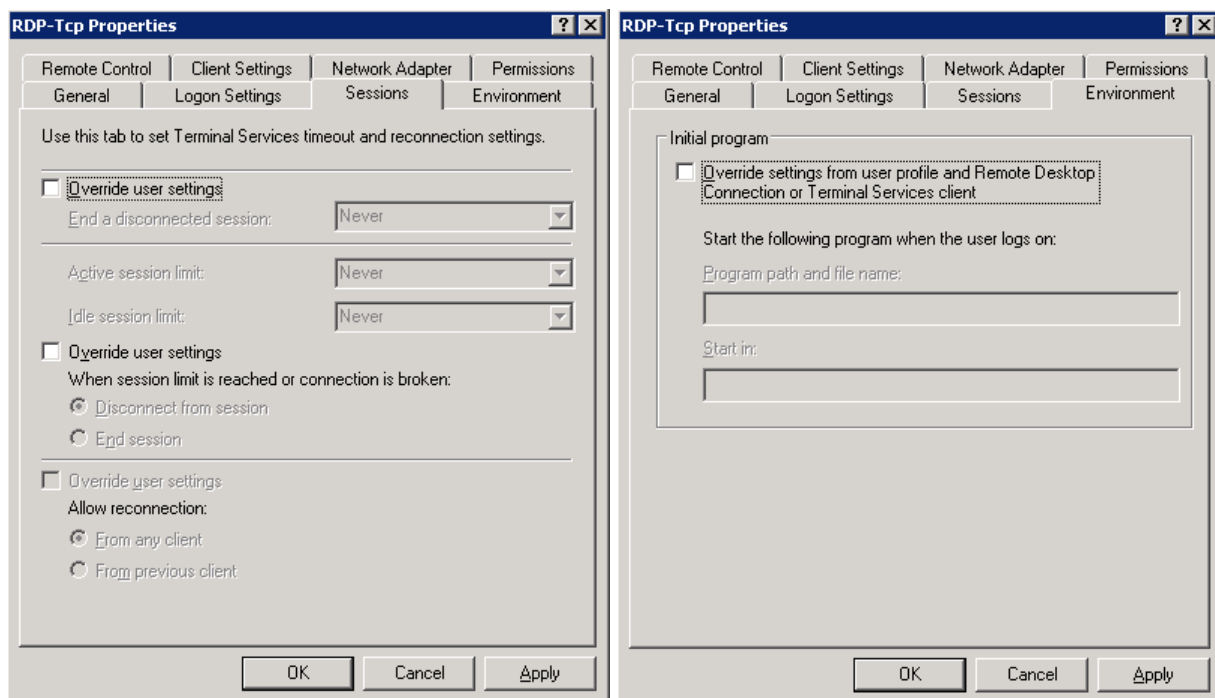


Figure 2.10: The Sessions and Environment tabs of the RDP connection properties.

The next tabs we'll explore are Remote Control and Client Settings (see Figure 2.11). These control shadowing and client resource redirection, respectively. Once again, these settings inherit their behavior from the user account or Remote Desktop Connection software by default.

When an administrator wants to remotely connect to an existing user's session to provide support, this action is called *shadowing* or *remote control*. On the Remote Control tab, you can keep the default setting of inheriting shadowing settings from the user's account attributes, or you can specify your own for this server. If you specify settings here, your options are to enable or disable the requirement for the user to give permission before being shadowed (via a popup window) and to control the level of interaction that the administrator can have with the user's session—either view only or interact. If you select interact with the user's session, the administrator will be able to control the user's mouse and enter keystrokes on behalf of the user. You also have the option to disable remote control altogether.



Before disabling the *Require the user's permission* setting, be sure to confirm that you are not under a legal obligation to inform users when they are being shadowed. Many states and industries require this communication with users.

The Client Settings tab lets you override the client resource redirection settings specified in the Remote Desktop Connection client software. On this tab, you can enable or disable the redirection of the following client resources:

- Drives
- Printers
- LPT ports
- COM ports
- Clipboard
- Audio

You can also specify whether to default to the main client printer and limit the maximum color depth that a user can request when connecting to the server. Higher color depths can degrade performance over slow connections.

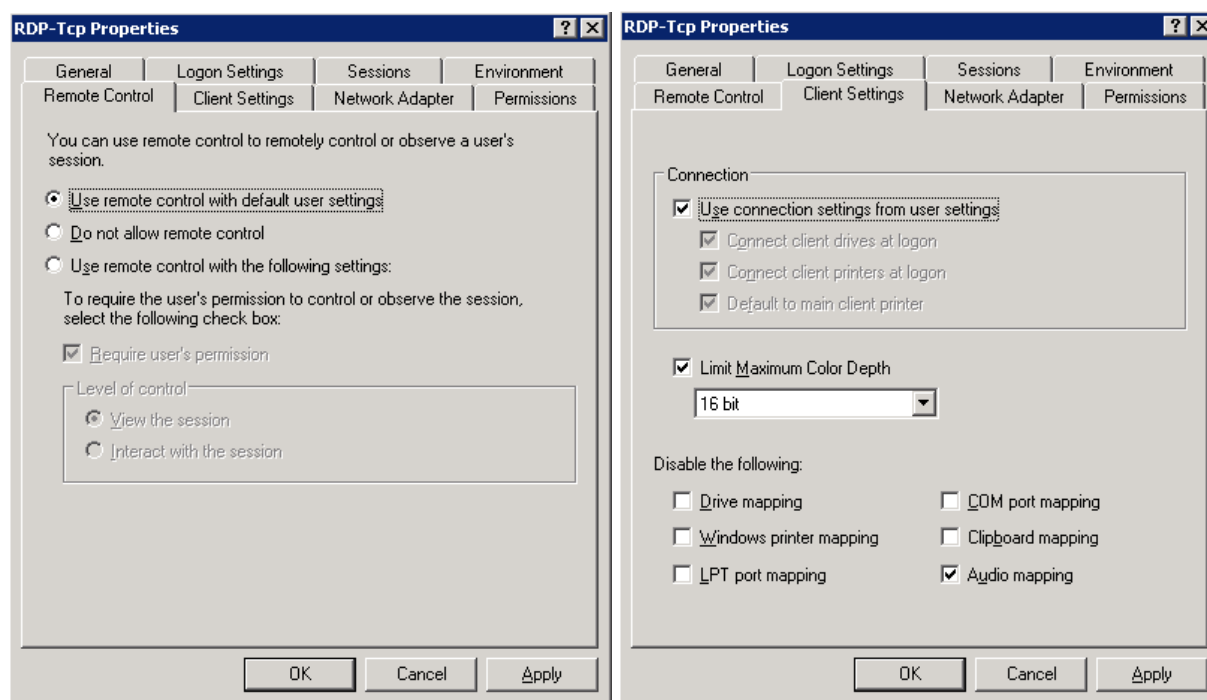


Figure 2.11: The Remote Control and Client Settings tabs of the RDP connection properties.

The Network Adapter and Permissions tabs are dedicated to more server-centric settings (see Figure 2.12). The Network Adapter tab lets you specify whether this set of connection settings applies to all network adapters or, in the case of a multi-homed server, a specific adapter. The Permissions tab is where you control who has the ability to connect to the server using RDP, and what level of rights they have when it comes to accessing virtual channels interacting with other sessions on the server. Figure 2.12 shows both of these tabs.

In addition to limiting the RDP connection to one network interface, the Network Adapter tab lets you set a limit on the total number of connections allowed on the specified interface or on the entire server if *All network adapters configured with this protocol* is selected. If you have more than one network adapter in your server, and you select a specific adapter on this tab, you will be able to create a new connection in the main Terminal Services Configuration window and apply separate settings to it.

The Permissions tab is one in which major change has occurred since Win2K. Under Win2K, the default permissions allowed all users from any trusted domain to immediately connect to the server once Terminal Services was enabled. WS2K3 is secure by default and only allows administrators and members of the Remote Desktop Users group to connect. Keep in mind that the Remote Desktop Users group is empty by default, so in order for users to connect to your terminal server, you will need to add them to this group.

👉 If you are in an AD domain, you can use the Managed Group setting in Group Policy to control the members of the Remote Desktop Users group.

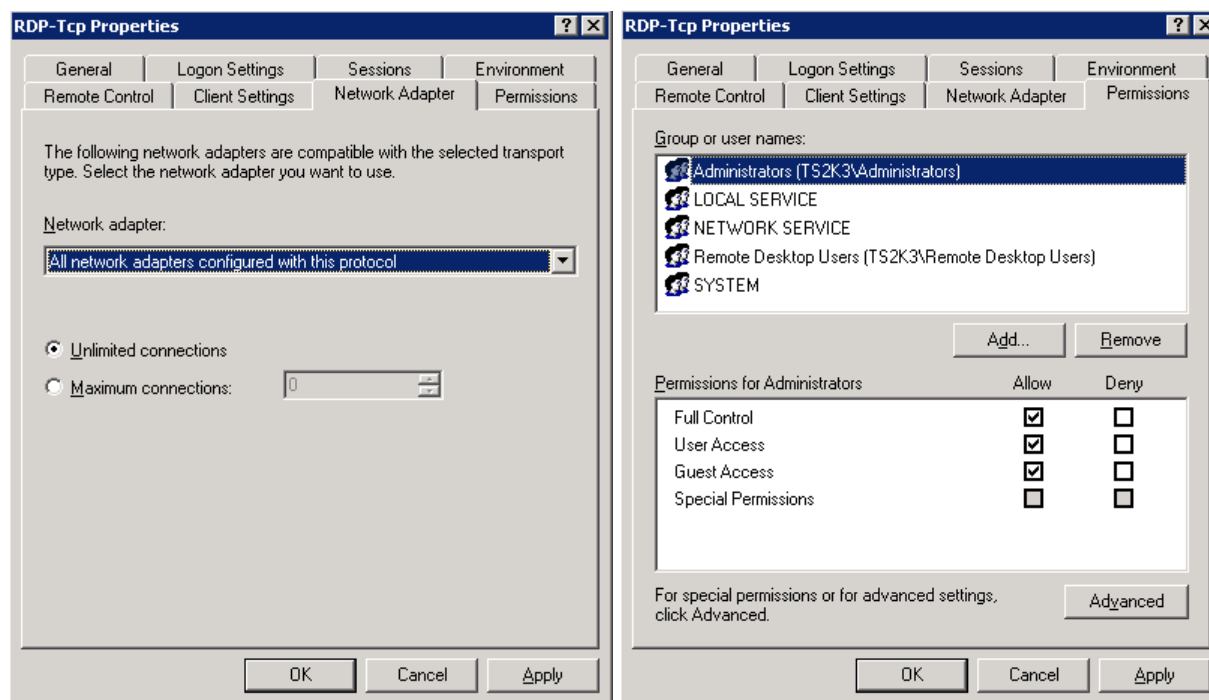



Figure 2.12: The Network Adapter and Permissions tabs of the RDP connection properties.

Group Policy–Based Configuration

WS2K3 has exposed a large number of settings to the Group Policy editor that were not available under Win2K. If your terminal servers are in an AD environment, you will definitely want to take advantage of Group Policies; but even in a workgroup or NT 4.0 domain environment, the terminal server settings are still available to you via the local machine policy. In this section, I'll go over some of the settings available in the Group Policy editor.

 In Chapter 4, I'll cover some advanced techniques available in an AD environment.

To access the local Group Policy editor, launch GPEDIT.MSC from a run command or command line. Navigate to the Terminal Services settings node in Computer Configuration, Administrative Templates, Windows Components, Terminal Services. As you can see in Figure 2.13, there are many settings available.

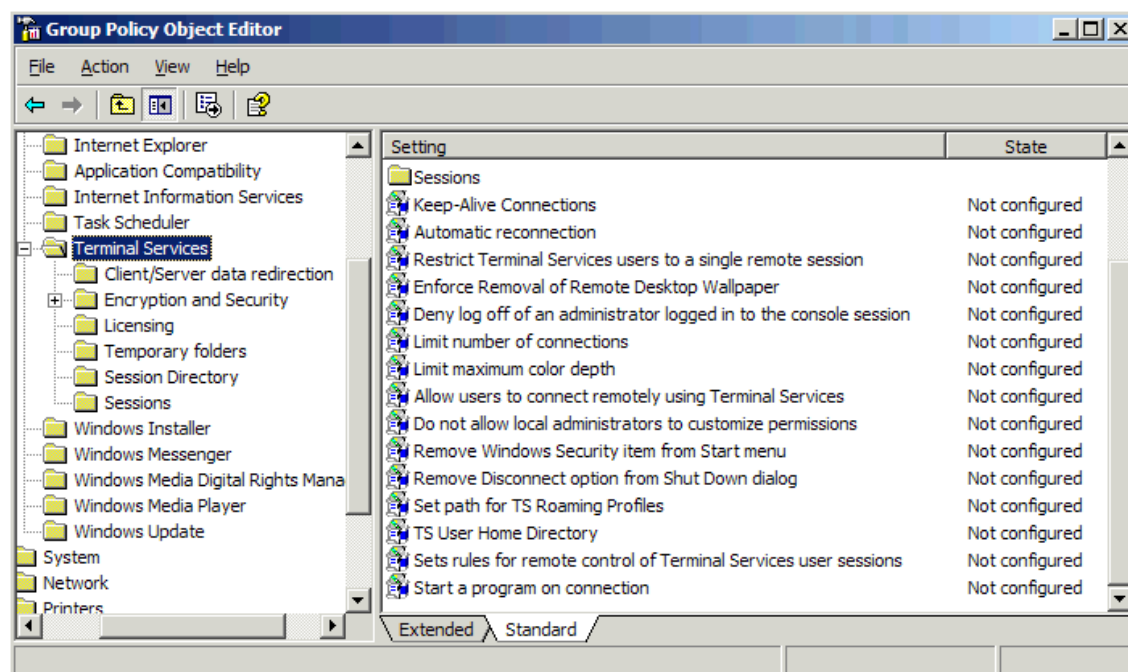


Figure 2.13: Terminal Services settings in the Group Policy Object Editor.

Settings are divided into categories: Encryption, Licensing, Sessions, and so on. You'll notice that several settings are identical to those found in the Terminal Services Configuration tool. The reason is so you can centrally manage settings on your servers without having to configure each server manually. Some settings to take a close look at are:


- **Set path for TS Roaming Profiles**—This setting lets you specify a server and share in which to store terminal server user profiles. You can also specify a TS profile path on each user's account. This setting not only lets you override the setting in the user account on a per server basis, but also enables you to specify a different terminal server profile server for groups of terminal servers. This ability is helpful if you have geographically dispersed server farms and users that roam between them.
- **TS User Home Directory**—This setting is similar to the previous setting except that this one specifies a server and share to create home directories for users logging on to the terminal server.

☞ When configuring either of the previous settings, do not attempt to specify a per-user subdirectory. The server will automatically append %username% to the path.

- **Do not allow local administrators to customize permissions**—This setting disables the Permissions tab in the Terminal Services Configuration tool. Because WS2K3's RDP is restricted by default and the preferred method of granting users access to the terminal server is by adding them to the Remote Desktop Users group (as opposed to adding new groups to the RDP permissions), you can now disable the tab entirely.

The licensing node under Terminal Services is used to configure a terminal server license server by enforcing a license server security group or disabling license upgrades. A license server security group restricts the license server to only issuing tokens to servers that are members of the Terminal Services Computers security group. Disabling license upgrades prevents the license server from issuing WS2K3 terminal server tokens to clients connecting to a Win2K terminal server. By default, if the license server has no Win2K terminal server tokens available, it will issue WS2K3 tokens instead.


The Session Directory node is used to configure terminal servers that are members of a Session Directory cluster. Through this node, you can specify the name of the cluster and the Session Directory server and control the behavior when clients attempt to reconnect to an existing session on the cluster.

 I will cover Session Directory in depth in Chapter 3.

In addition to these settings, there are a few more settings of interest to terminal server administrators. Under Computer Configuration, Administrative Templates, System, User Profiles, there is an option to *Allow only local user profiles*. This setting prevents a server from downloading roaming user profiles, even if one is configured on the user's account. This setting comes in handy if you have a terminal server at a separate site than your profile server and you do not want to establish a separate profile server for the site. By enabling this policy, when a user logs onto the server, a local profile is created and stored on the server.

The User Profiles node also holds the *Delete cached copies of roaming profiles* setting. This setting instructs the server to purge the local copy of a profile when a user logs off (if the user has a roaming profile). This setting lets you save disk space on your terminal server and prevents old versions of profiles from merging with the network copy if a user hasn't logged onto a particular terminal server in a while.

If you look under User Configuration, Administrative Templates, Windows Components, Terminal Services, you will see that the settings for remote control, environment, and session timeouts are available here in addition to the Computer Configuration node. This dual accessibility allows you to configure these settings on a per-user basis if you choose.

 In most cases, if you configure the same setting in both Computer Configuration and User Configuration, the machine setting wins.

Additional Configuration Settings

In addition to the settings available in the Terminal Services Configuration tool and the Group Policy Object Editor, there are several settings that most Terminal Services administrators control via manual registry edits. These settings let you improve performance on your servers by increasing the number of idle RDP sessions and disabling display features to minimize screen redraws:

- **Idle RDP connections**—By default, the server creates two idle RDP sessions to respond when a client opens a connection. These sessions are immediately replaced with a new idle session when a user connects, but to prevent the rare case when more than two connections are established at the exact same moment, you can increase the number of idle sessions. I recommend increasing it to five by setting the `IdleWinStationPoolCount` value of the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server` subkey to 5.
- **User overrides for desktop settings**—These settings override the user's preferences when logging on over RDP to optimize performance and reduce screen refreshes. To disable animation when resizing windows, set the `MinAnimate` value in the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\UserOverride\Control Panel\Desktop\WindowMetrics` subkey to 0. In addition, you can set the following values (defined in Table 2.1) of the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\UserOverride\Control Panel\Desktop` subkey.

Value	Setting	Description
AutoEndTasks	1	Automatically terminates programs that aren't responding
CursorBlinkRate	-1	Prevents the cursor from blinking, which cuts down on screen redraws
DragFullWindows	0	Doesn't show contents while dragging a window
MenuShowDelay	10	Sets the delay for showing submenus
WaitToKillAppTimeout	20000	The number of milliseconds to wait before terminating an application that isn't responding
SmoothScroll	Dword-type value of 00000000	Disables smooth scrolling
Wallpaper	(none)	Disables wallpaper

Table 2.1: Registry values to set user override settings.

In addition to these registry changes, you should change the following settings to tune the overall server performance:

- **Tune the event logs**—In Event Viewer, adjust the properties of each log so that its maximum size is 1MB (or larger if you want an extended history), and set it to overwrite as needed. These settings can also be controlled in a domain Group Policy under Computer Configuration, Security Settings.
- **Configure the capture of debugging information**—In the Advanced Startup and Recovery Options of the System Control Panel applet, adjust the Write Debugging Information settings to save the debug dump to an appropriate location (or disable it entirely), and confirm that the server is set to automatically reboot.

- Examine the Run registry key—Under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run, there can be values pointing to executables to launch at the start of each session. Some applications (including certain virus scanners, network teaming, and load balancing utilities) add entries here to create system tray icons. Usually these icons only provide easy access to control panel or configuration utilities and do not really need to be run at all times. By removing these entries, you can reduce the overhead of having these applets run in every terminal server session. Be sure to check with the application vendor before removing.

Installing and Configuring the Remote Desktop Connection Client

After you have installed and configured a terminal server, you will want to provide access to the server via a client interface. There are two versions of the Remote Desktop Connection client: the local version, available for Windows 32-bit OSs, Macintosh, and PocketPC; and the Remote Desktop Web Connection, an ActiveX control used to connect to a terminal server through an Internet Explorer (IE) window.

Which client you use depends on your needs. If you want to maintain the configuration of terminal server connections on the clients (by distributing RDP files or letting users manually enter server names), the Remote Desktop Connection client is perfect. If, however, you want to centrally control the connection (server names, initial programs, experience options), the Remote Desktop Web Connection might be a better option.

Remote Desktop Connection Client

The Remote Desktop Connection client comes pre-installed on all WindowsXP computers, as this is the same client used for Windows XP's Remote Desktop feature. If you want to install or deploy it on other OSs, you can download it from

<http://www.microsoft.com/windowsserver2003/technologies/terminalservices/default.msp>. The client can be installed on the following OSs: Windows 95, Windows 98, Windows 98 Second Edition, Windows Me, Windows NT 4.0, Win2K, and Macintosh. You can also download the Terminal Services Client for PocketPC from

<http://www.microsoft.com/mobile/pocketpc/downloads/default.asp>. Figure 2.14 shows the Remote Desktop Connection client's interface.

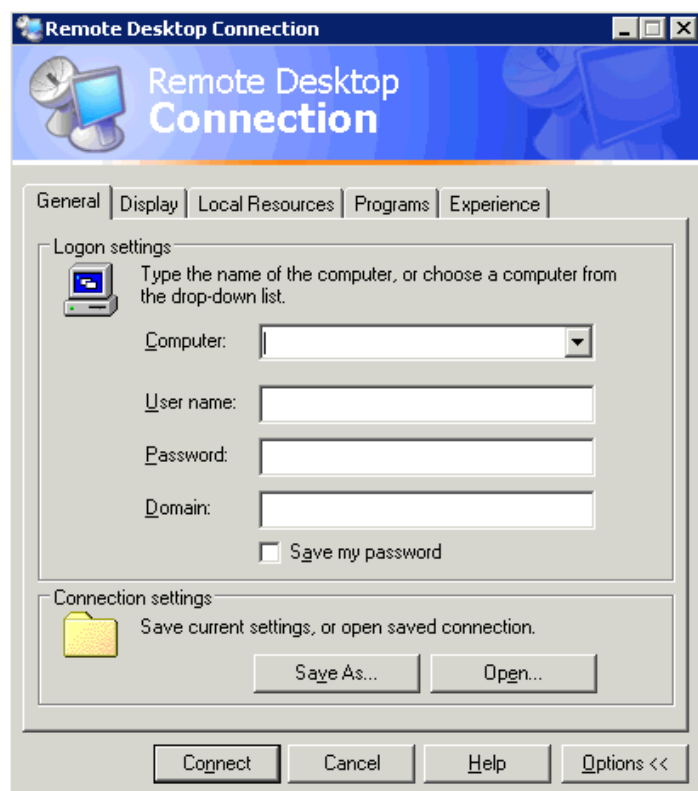


Figure 2.14: The Remote Desktop Connection client.

In the client's interface you can set the following options:

- Name or IP address of the terminal server to connect to
- Username and password to use when connecting
- Screen size
- Color depth
- Sound, local drive, printer, and COM port mapping
- Behavior of Windows key combinations (interpreted by the client or by the server)
- An initial program to run in lieu of a desktop
- Experience options (enable or disable certain visual effects to improve performance over slow connections)

Once you set your options the way you like, you can save the configuration to an RDP file. This file is a text-based file that can be executed for easy connection to a specific server or application. Administrators can pre-create RDP files for users and distribute the files via email.



If you copy an RDP file with a saved password to another computer, the password will not be entered upon connection. This behavior is important to know if you intend to distribute RDP files with credentials included.

If an administrator has configured any of the options in the Terminal Services Configuration tool and the Group Policy Object Editor that correlate to settings available in the client, the server settings override those requested by the client. Settings in Group Policy take precedence over both the Terminal Services Configuration tool and the client options.

You can also control the Remote Desktop Connection client via the command line. Listing 2.1 shows the syntax.

```
MSTSC [<Connection File>][/v:<server[:port]>] [/console]
[[/f[fullscreen]|[/w:<width> /h:<height>]]|[/Edit"connection
file"][/Migrate]
```

<Connection File> - specified the RDP file for the connection
 /v:<server[:port]> - specifies the server name or IP address to connect to and the port on which to connect
 /console - connect to the console session of a Windows Server 2003
 /f[fullscreen] - starts the client in full screen mode
 /w:<width> /h:<height> - specifies a height and width for the connection window
 /edit - opens an RDP file for editing
 /Migrate - migrates legacy Client Connection Manager connections out of the registry and into RDP files.

Listing 2.1: Syntax for configuring the Remote Desktop Connection client via the command line.



The command line is the only way to connect to the console session of WS2K3 using the Remote Desktop Connection client. There is a GUI option for this in the Remote Desktops Administrative Tool on WS2K3 or in the WS2K3 Administrative Tools package available for installation on Windows XP.

Remote Desktop Web Connection

The Remote Desktop Web Connection installs on an Internet Information Server (IIS—or a WS2K3 system that has the application server role enabled). The Remote Desktop Web Connection package is available from Microsoft for installation on Win2K IIS or can be installed on WS2K3 by selecting Add/Remove Windows Components, Application Server (details), IIS (details), World Wide Web Service (details), Remote Desktop Web Connection. The former installs in C:\inetpub\wwwroot\tsweb by default, and the latter installs in C:\windows\web\tsweb.

As you can see in Figure 2.15, the only options available in the default interface are Server name, window size, and logon information (username and password). The Remote Desktop Web Connection's ActiveX control actually supports the full range of options available in the local client, you just need a little programming skill to take advantage of them. The interfaces available to the control can be found by searching <http://MSDN.Microsoft.com> for Remote Desktop ActiveX Control Interfaces.



In the Appendix, I'll offer an example of how to add an Experience drop-down box to the default page.



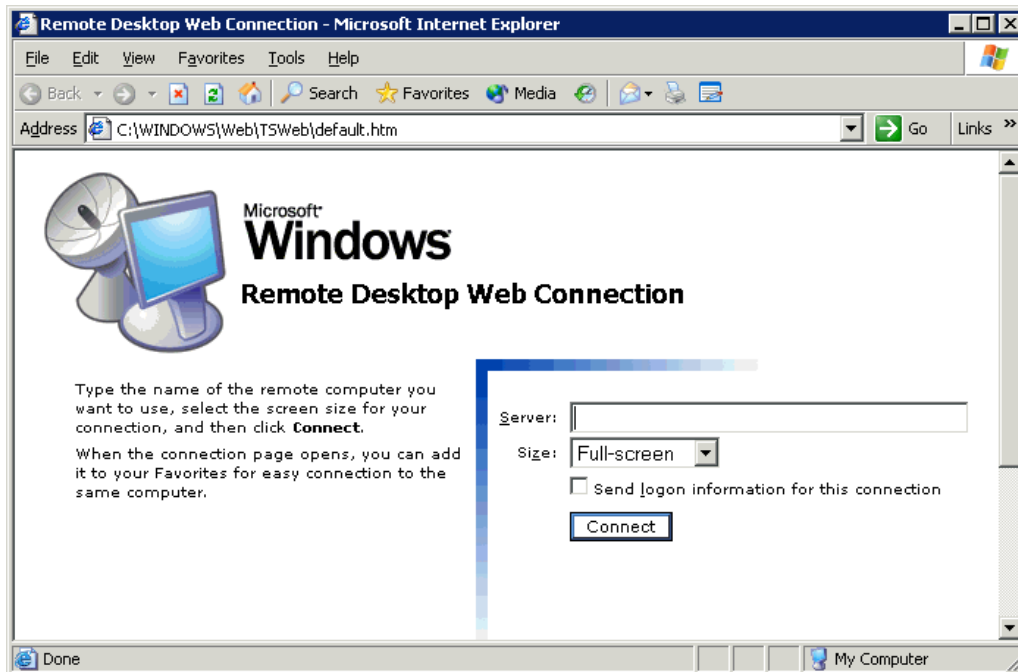


Figure 2.15: The Remote Desktop Web Connection interface.

Remote Desktops Administrative Tool

In addition to the two user-focused clients, WS2K3 includes a new version of the Terminal Services Connections tool from Win2K. The new tool is called simply Remote Desktops, and can be found in the Start menu under Administrative Tools. You can also install this tool on WindowsXP by using the adminpak.msi package found on the WS2K3 source CD-ROM.

Figure 2.16 shows the Remote Desktops tool with a number of connections defined. By using this tool, an administrator can store connection definitions to all servers in one interface. The tool can even be used to connect to Win2K servers in Remote Administration or Application Server mode or even WindowsXP desktops with Remote Desktop enabled.

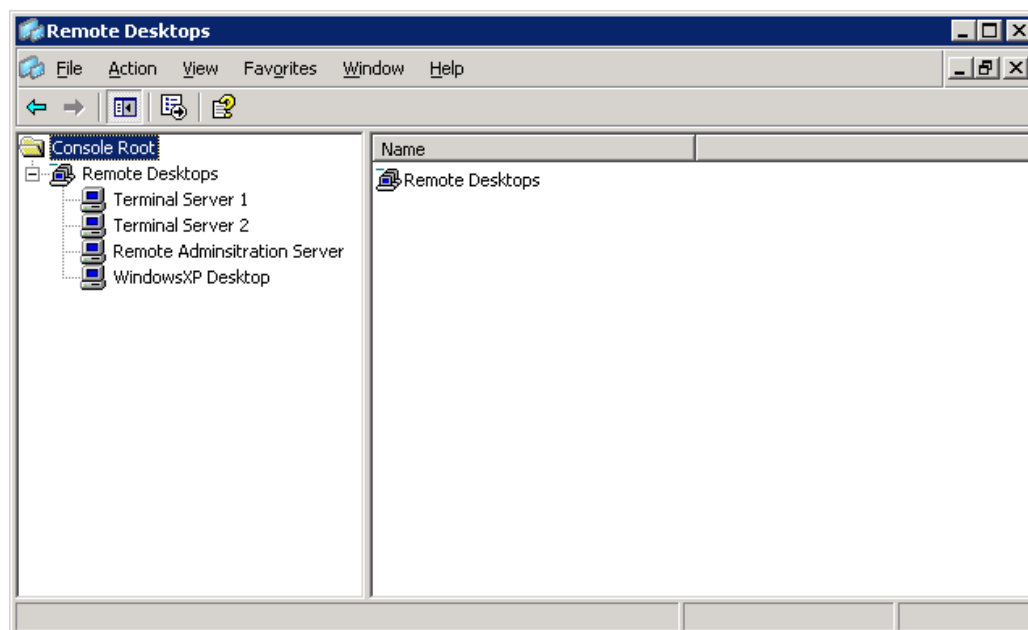


Figure 2.16: The Remote Desktops administrative tool.

When you select one of the defined connections, the desktop of the server is displayed in the right pane. You can establish connections to multiple servers and toggle between them by selecting their icons on the left pane. The tool will even allow you to store credentials or select a program to run in lieu of the desktop.

☞ If you prefer to work with your terminal server connections in separate windows, you can mimic the function of the Remote Desktops tool by creating a folder containing all of your RDP files and create a shortcut to the folder on your desktop or Start menu.

Summary

In this chapter, I walked you through the process of installing and configuring the terminal server role. I gave you an overview of the new settings available in the Terminal Services Configuration and Group Policy editor tools. Next, I explored the clients available to connect with terminal servers.

In Chapter 3, I will introduce you to a new load-balancing technology available in WS2K3—Session Directory. I will also discuss other load-balancing options and talk about server sizing and capacity. Finally, we will briefly explore a new concept in terminal server design: using virtual machines to host terminal servers.