Realtime
publishers
"Leading the Conversation"

The Definitive Guide™ To

Building a Windows Server 2008 Infrastructure

sponsored by

triCerat

Greg Shields

Copyright Statement

Chapter 7: Introduction to Terminal Services

At the beginning of the previous chapter on Group Policy, we talked about how the world of computing has evolved from its early focus on relatively “dumb” consoles connecting to mainframe computers back in the data center. These days, a large percentage of business computing is done using the client/server model, where clients and servers both handle some component of workload processing. Servers sit back in the data center and typically handle the processing of a single function or service, while powerful desktops are used in concert to locally accomplish much of the workload processing.

There is great power in this division of processing between client and server. The actual work involved in processing users’ data needs is distributed among dozens to hundreds of processors within many machines, rather than consuming the resources of a smaller number of total processors within a much smaller collection of computing platforms. Users are given greater freedoms with the types of workloads they can accomplish. And adding new applications or services imposes a comparatively smaller impact to the greater user environment.

But with this greater distribution of workload processing comes a related explosion of touch points for administration and security. With the client/server computing model, each server and client runs an OS all its own, with all the associated management requirements. Administering such an IT environment today means controlling the configuration of computers in both the data center and at the desktop, which is a bigger scope to wrap your arms around. Also problematic are applications that require “chatty” network conversations between the client and server halves. These network conversations are particularly talkative, which increases the level of networking connectivity required between client and server. As clients that use such applications move farther away from their respective servers, overall performance declines, particularly as the number of WAN links involved increases.

One solution that brings together the power of client/server processing with the improved security, administration, and networking architectures of the mainframe days is Terminal Services. With Windows Server 2008, Terminal Services arrives as an installable Role that enables users to share the use of a server-class system in much the same way as in the old mainframe and terminal days. By installing applications and pointing users to the Terminal Server, users gain remote access to applications while administrators gain substantial management benefits.

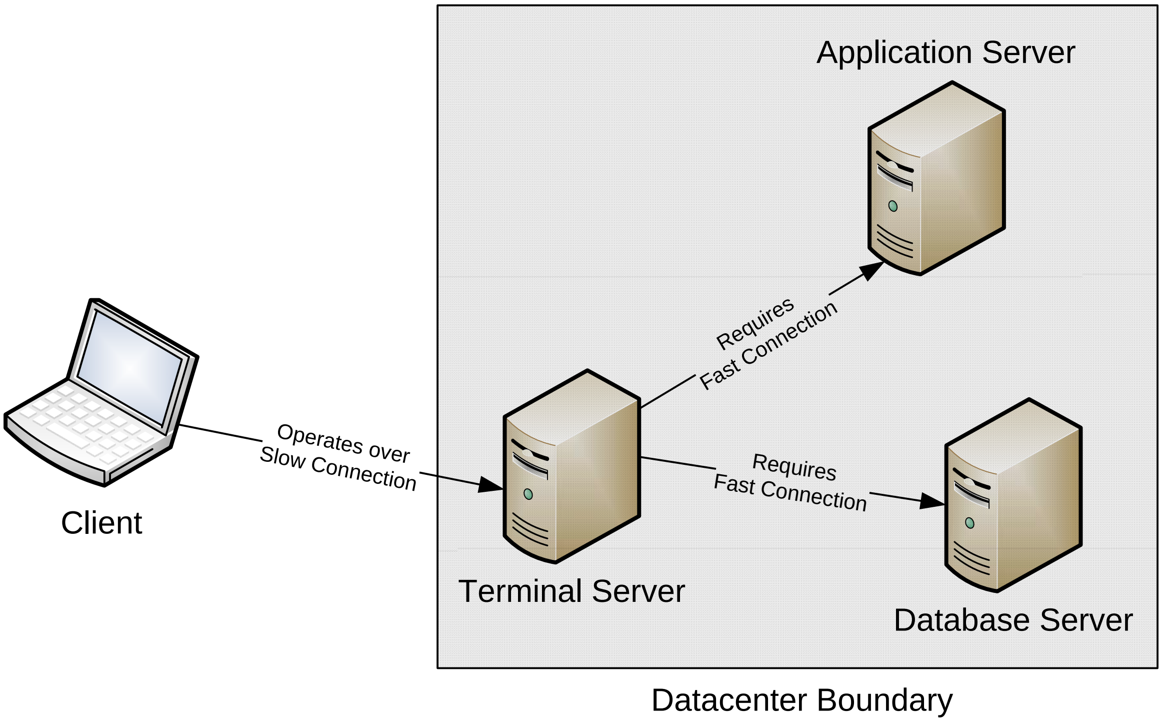


Figure 7.1: Terminal Services enables an interface for clients on slow connections to access applications as if they were within the high-speed data center boundary. Moving client applications closer to their respective servers results in improved application performance.

But easier management isn’t the only reason to move applications to Terminal Services. Making this solution even more useful to IT is the nature of the network protocol used by Terminal Services itself. The Remote Desktop Protocol (RDP) used to connect clients to Terminal Services servers is designed to consume an extremely small amount of bandwidth. Thus, the same Terminal Server connected to by clients from the local network over high-speed links can also be connected to by clients in remote locations over slow connections. RDP is extremely tolerant of latent and low-bandwidth network conditions, making it an excellent solution for enabling far-reaching access to applications.

What Exactly Is Terminal Services?

To put it bluntly, Terminal Services effectively turns a Windows server into a giant Windows workstation, the use of which is shared by each connected user. Unlike virtually all other servers in the data center, the installation of Terminal Services enables non-administrative users to access a server session in the same way they would their own desktop. On that Windows server are installed the applications and other resources required by its users. Users are given a client, the Remote Desktop Client (RDC), with which to enable the connection to the desktop of that Terminal Server.

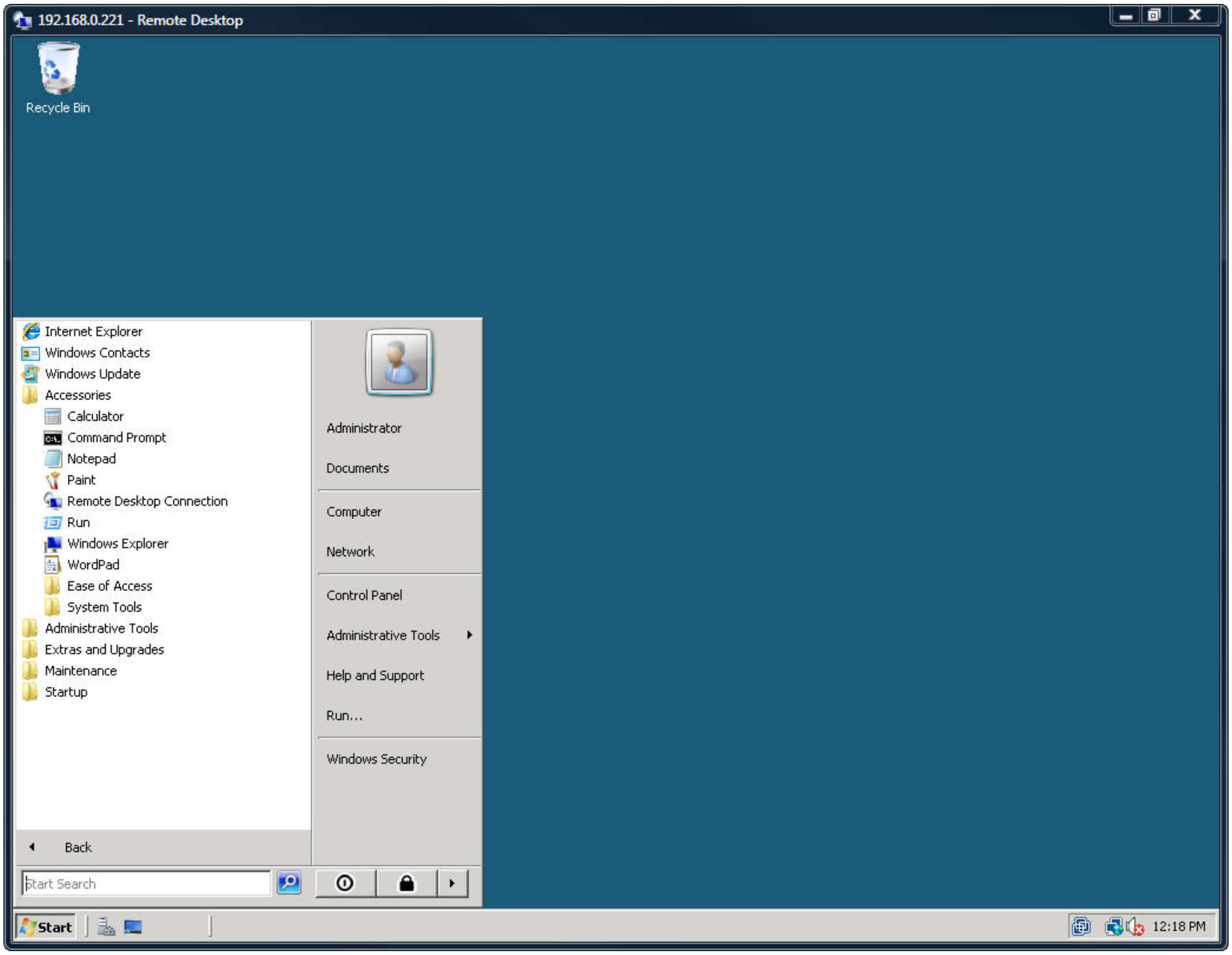


Figure 7.2: An example of a Terminal Services-hosted desktop connected using the RDC.

The Windows OS has the native capability to operate multiple “sessions” on the same server. Each session hosts the desktop and operating environment for a single user. The first of these, the “console” session, is used when an administrator is actively working at the console of the server. Additional sessions are created when users connect via the RDC. Sessions are administratively separate from each other, containing their own processes and process threads. Applications installed on the server are shared by all sessions, with the processes associated with each application being invoked individually into each session.

Applications hosted on Terminal Services in certain circumstances have the ability to share available resources between session processes. However, this process depends on how the application was coded and the rate at which DLLs are modified during use. External third-party tools are often necessary to ensure that resource sharing is being accomplished at optimum levels.

The end result for the client is a connection either to a Terminal Server desktop, as shown in Figure 7.2, or one of its individual applications. The user interacts with applications on their local desktop as if they were running locally, while in fact those applications are actually being processed by the Terminal Server. The RDC processes little more than screen updates and mouse and keyboard commands between the server and the client for user visualization. Due to compression on the part of the protocol, screen updates require very little bandwidth to maintain an acceptable user experience and are relatively tolerant to network latency.

> ✎ With Terminal Services, you'll often hear about the concept of *user experience*. This is a key point of Terminal Server administration. With large numbers of users simultaneously sharing resources on the same server, one of the roles of the Terminal Server administrator is to play "systems babysitter," ensuring that users are not consuming more than their fair share of resources. The goal of this administrative activity is to ensure that the *user's experience* with the Terminal Server-hosted application is equivalent or better than what they would have experienced in a local installation of the application.
>
> You'll find that managing that experience is one of the biggest jobs of the Terminal Server administrator, especially as the count of Terminal Servers increases. Tools both native to Terminal Server as well as available through third parties are available to assist with the management of this experience.

With this understanding of what Terminal Services is, it is important to also understand the benefits it can provide. By leveraging the Terminal Services Role atop a Windows Server 2008 computer, the IT environment gains a number of benefits to administration and workload processing:

- *Centralization of Management.* In Chapter 6, we talked about Group Policy and how Active Directory's Group Policy enables administrators to control large numbers of computers through a single configuration policy. Group Policy is effective for accomplishing this task, but its use still incurs an administrative cost for managing the configuration of each desktop's settings. By moving applications from the desktop to shared Terminal Servers, this reduces the total count of points of configuration. Fewer points of configuration mean fewer places where mistakes and omissions can be made, and an overall reduction in the cost to administer the environment.
- *Centralization of Applications.* When applications are distributed to individual desktops, this incurs a management cost associated with their administration. That cost relates to the time required to install the application, updates that are later required, as well as the cost to physically travel to the local machine when troubleshooting support is required. By moving applications from the desktop to shared Terminal Servers, large numbers of users are effectively sharing the use of a smaller count of application instances. Although this does not necessarily reduce the licensing cost for those applications, it does reduce the number of touch points for managing those applications. Application updates are similarly affected because fewer application instances ultimately require fewer updates.
- *Centralization of Security*. Ensuring the security of a distributed environment can also be a management headache due simply to the large number of areas that require securing. Applications in addition to OSs require controls in place to maintain the security of data and processing. By moving applications from the desktop to shared Terminal Servers, the sheer number of points required to secure is reduced.

- *Proximity of Desktop Applications to Servers.* Some desktop applications require a large amount of communication to occur between the client application and the server. When clients running these applications are far removed from their servers, the end result is a reduction in performance for the application. Moving these applications from the desktop to shared Terminal Servers that are close in network proximity to their application servers results in an increase in application performance. A picture of this is shown in Figure 7.1. This increase occurs because the "chatty" client application is now within the high-speed data center boundary. The lightweight RDP is then used to pass screen updates and keyboard and mouse commands between client and Terminal Server while the "chatty" communication between Terminal Server and application server stays within the high-speed data center boundary.
- *Internet-based Application Hosting.* Since the network is no longer the bottleneck between applications and their back-end servers, it is similarly possible to enable the hosting of otherwise internal applications over the Internet. With the right authentication and encryption technology in place, which is natively available within Terminal Services, this allows the IT organization to securely extend the reach of critical corporate applications to virtually everywhere with an Internet connection.

Introducing Windows Server 2008's Terminal Services Role

With the release of Windows Server 2008, all the functionality commonly associated with Terminal Services has been encapsulated into an installable Role. Like the other Roles we've discussed in this guide to this point, the Terminal Services Role contains a set of Role Services that support its functionality. The benefit of this movement is that Role Services in support of Terminal Services can be installed without needing to install the core Terminal Services components.

The Terminal Services Role in Server 2008 itself enables effectively no functionality. Installing the Role minimally requires the installation of at least one of the five Role Services enumerated in the subheads that follow. For smaller installations, you might find multiple Role Services installed onto a single server. For larger environments, splitting the processing of these Role Services onto separate servers enables administrative separation as well as better security for each component.

Server

The Terminal Server Role Service is the component most commonly associated with Terminal Services. Installing this Role Service enables the server to operate as a Terminal Server, accepting inbound RDP clients and handling multiple session creation. Due to how multiple sessioning impacts applications installed on the server, it is a best practice to install the Terminal Server Role Service first, prior to the installation of any applications. As we'll discuss later, applications installed to Terminal Services must be installed using a special server mode for them to function properly.

TS Licensing

Clients that make use of Terminal Services require a special kind of license called a Terminal Server Client Access License (TS CAL). This special and additional license can be distributed on a per-user or per-device basis. Management of those licenses is handled by the TS Licensing Role Service. At a minimum, one instance of TS Licensing must be present within the Active Directory Forest before clients will be allowed to connect to the Terminal Server.

TS Web Access

As we'll discuss later on in this chapter, once Terminal Services is installed to a server there are multiple mechanisms that can be provided to connect users to hosted applications. One mechanism involves pointing users to a Web site where links to hosted applications are made available. The TS Web Access Role Service is the built-in mechanism for hosting this Web site. TS Web Access interfaces with configured applications on a Terminal Server to provide a friendly Web-based interface for users.

TS Gateway

By default, the RDP network protocol passes data across the network in an unencrypted format. For internal-only or low-risk environments, this is often an acceptable solution. But some environments require higher levels of security for this data. The TS Gateway Role Service enables IPSec-based transport-level authentication and encryption for RDP network traffic. As a network gateway, the TS Gateway Role Service also serves as a type of proxy server, proxying traffic between external clients and Terminal Services. This function serves to obfuscate internal services and is an excellent solution for hosting applications over the Internet.

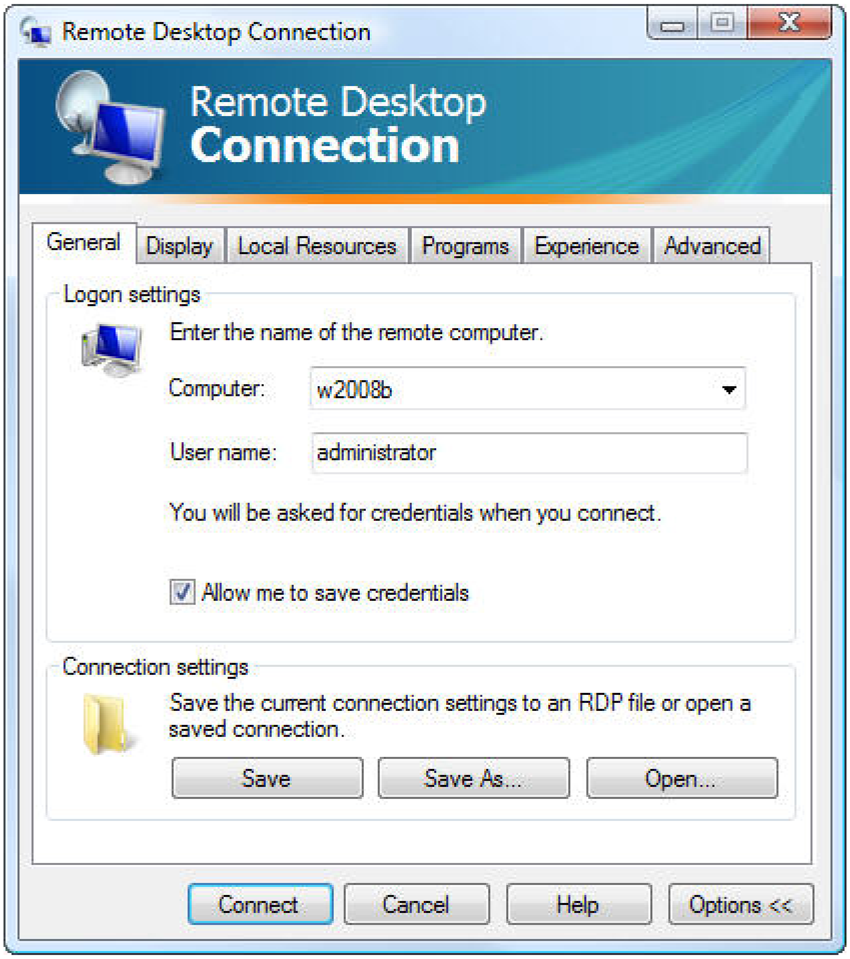
TS Session Broker

Organizations that invest in Terminal Services for hosting of applications often require high-availability support to ensure that the loss of a single server will not mean significant downtime of the application. TS Session Broker is a built-in load-balancing and high-availability tool that allows clients to connect to multiple back-end Terminal Servers as if they were a single entity. This capability provides a single point of contact for multiple servers.

> ✎ In this chapter, we'll talk about the client side of Terminal Services as well as the first two of these Role Services. In Chapter 8, we'll continue the discussion with a detailed explanation of the use and utility of the other three Role Services as well as some best practices associated with the use of Terminal Services.

The Remote Desktop Client

In addition to the server-side components discussed earlier, individual clients require the installation and use of the RDC to connect to Terminal Servers. The RDC is a small tool that enables a client to connect to the Terminal Server. It handles receiving screen updates from the server while sending keyboard and mouse commands back to that server. The RDC can either be run interactively (the console is shown in Figure 7.3) or it can be run in the background. Depending on how you plan to distribute links to hosted applications to your users, those users may actively use its administrative interface to connect to servers and applications or they may click links on Web pages or installed to their desktops to launch the client in the background.



Enter the name of the remote computer.

Computer: w2008b

User name: administrator

You will be asked for credentials when you connect.

☑ Allow me to save credentials

Save the current connection settings to an RDP file or open a saved connection.

Figure 7.3: The RDC tool can be run either interactively or in the background. When run interactively, a number of configuration settings can be made by users as they initiate their connection to a Terminal Server.

Running the client interactively enables a user to connect directly to the desktop of a Terminal Server. As you can see in Figure 7.3, to connect to the desktop of the server *w2008b*, the user needs only to launch the client and enter that server's name in the box next to *Computer*. Credentials can be associated with a server connection if desired. By clicking through the tabs at the top of the console, the user can configure elements of their user experience such as display size, color depth, local resources that are connected into the remote session, experience elements that add or remove graphical features from the session, and advanced functionality such as server authentication and TS Gateway settings.

Although using the RDC in interactive mode is useful for connecting directly to a Terminal Server's desktop, it is not possible through this interface to access an individually-hosted application. These hosted applications, called *TS RemoteApps*, allow the user to interact with a single application rather than an entire desktop. We'll talk in the next chapter about the functionality and benefits of TS RemoteApps, but know that connecting directly to a TS RemoteApp requires the use of a link either on the user's computer, hosted via a file share, or through a TS Web Access site.

✎ For the purposes of the functionality enabled with Windows Server 2008 and for our discussion in this and the next chapter, the minimum RDC version should be RDC v6.1. At the time of this writing, RDC v6.1 is available through the installation of Windows XP Service Pack 3 or Windows Vista Service Pack 1, and is available as a separate download for Windows XP Service Pack 2. RDC v6.1 is natively available on Windows Server 2008 RTM.

📖 More information about the features available in the RDC v6.1 can be found at http://support.microsoft.com/kb/951616.

Installing the Terminal Server Role Service

Switching back to the server side, before any client can connect to a Terminal Server, we must first install the Terminal Server Role Service. Do this within Server Manager by installing the *Terminal Services* Role and then the *Terminal Server* Role Service. Server Manager will prompt you to make three configuration determinations as part of the installation:

- *Authentication method.* Network Level Authentication (NLA) is an enhanced security mechanism that requires clients to authenticate to the Terminal Server before they are given access to a session. The use of NLA requires support at both the client and the server, and is only supported on RDC versions v6.0 or later. If your clients are at this level, it is a good idea to set this to *Require Network Level Authentication*.
- *Licensing Mode.* Terminal Services licensing can be done either *Per Device* or *Per User*. The determination about which licensing type to use will be up to your unique environment conditions. As an example, if you have a small number of users that roam among a larger number of devices, then *Per User* can be a good selection. If the reverse is true, consider using *Per Device*. Should you have a general one-to-one mapping of users to devices—which is often the case in typical office environments—consider choosing the *Per User* mode. Be aware that the type of TS CALs purchased from Microsoft must match this selection.
- *User Groups.* Lastly, add the users or groups that have access to connect to this server. These can be individual users but are more often selected by Domain Global Group.

As stated earlier, it is critical that this Role Service is installed before any applications are installed to the server. Once the installation and reboot have completed, clients will immediately be able to connect to the server's desktop using the interactive mode explained previously. Figure 7.2 shows an example of a hosted desktop that has been connected to using this procedure.

Installing the TS Licensing Role Service

Once the Terminal Services Role Service is installed and its initial configuration has completed, clients will be immediately able to connect to sessions on the server. However, TS CALs are still required and a TS Licensing server must be available. Upon the installation of Terminal Services, each server automatically enjoys a 120-day grace period before this requirement is enforced by the server and unlicensed clients are prohibited from connecting. This grace period is put into place because the process of obtaining and installing TS CALs involves a separate purchase from Microsoft and subsequent installation of licenses to a TS Licensing server.

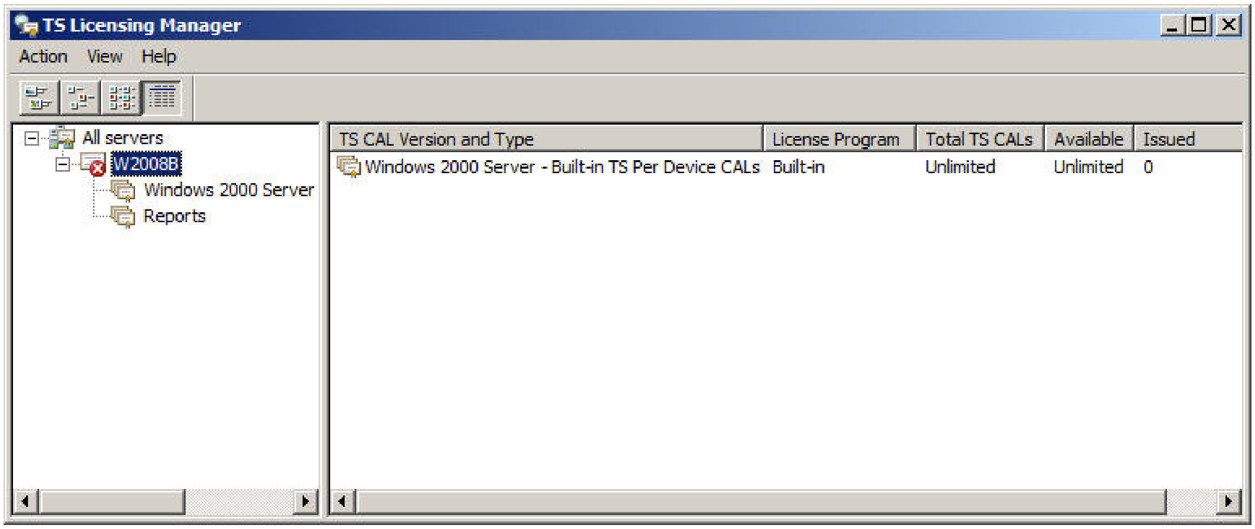
Assuming that the necessary TS CALs of the correct type (Per User vs. Per Device) have been purchased either directly from Microsoft or through a Microsoft partner, the first step in making available permanent licenses is to install the TS Licensing Role Service. Do this from Server Manager by right-clicking the *Terminal Services* node under *Role* and choosing to *Add Role Services*. Add the *TS Licensing* Role Service. Server Manager will prompt you to make one configuration decision as part of the installation:

- *TS Licensing Configuration.* Any TS Licensing server can be configured to serve licenses to its residing Workgroup, Domain, or Forest. Setting the discovery scope for the license server to one of these three values determines the boundary for serving licenses. Also asked here is the location for the TS Licensing database, which is a relatively small database installed local to the server.

Be aware of some idiosyncrasies associated with how the chosen licensing scope affects an individual Terminal Server's behavior in attempting to discover an available license server:

- The *Workgroup* licensing scope is only available when the computer is a member of a Windows Workgroup and not a member of a Domain. When the Workgroup licensing scope is enabled, Terminal Servers will automatically locate the TS Licensing server in their workgroup without additional configuration.
- When the *Domain* licensing scope is enabled, Terminal Servers will only be able to automatically locate the TS Licensing server when it is installed onto a Domain Controller. If TS Licensing is not installed onto a Domain Controller, each Terminal Server must be specifically configured to point to a license server. This can be done from within *Server Manager | Terminal Services | Terminal Services Configuration | Edit Settings | Licensing tab*. There, choose to *Use the specified license servers* and enter potential license server names separated by commas into the text box.
- When the *Forest* licensing scope is enabled, Terminal Servers will be able to automatically locate the TS Licensing Server without any additional configuration. This is because TS Licensing Servers that use the Forest licensing scope automatically publish information about their location into Active Directory. The installing administrator must have Enterprise Administrator privileges to accomplish this task.
- If TS Licensing is installed on the same server as Terminal Services, that server will be able to automatically locate the license server no matter what scope is selected.

Once TS Licensing is installed, the license server must be activated and licenses installed. To access the TS Licensing Manager console, navigate to *Administrative Tools | Terminal Services | TS Licensing Manager*. The resulting screen looks similar to Figure 7.4. Two steps are required to properly add licenses. First, right-click the server name and select *Activate Server* to launch the *Welcome to the Activate Server Wizard*. This wizard registers TS Licensing for this server with the Microsoft Clearinghouse, a process that can either be done over the Internet, via phone, or using a Web browser.



TS CAL Version and Type	License Program	Total TS CALs	Available	Issued
Windows 2000 Server - Built-in TS Per Device CALs	Built-in	Unlimited	Unlimited	0

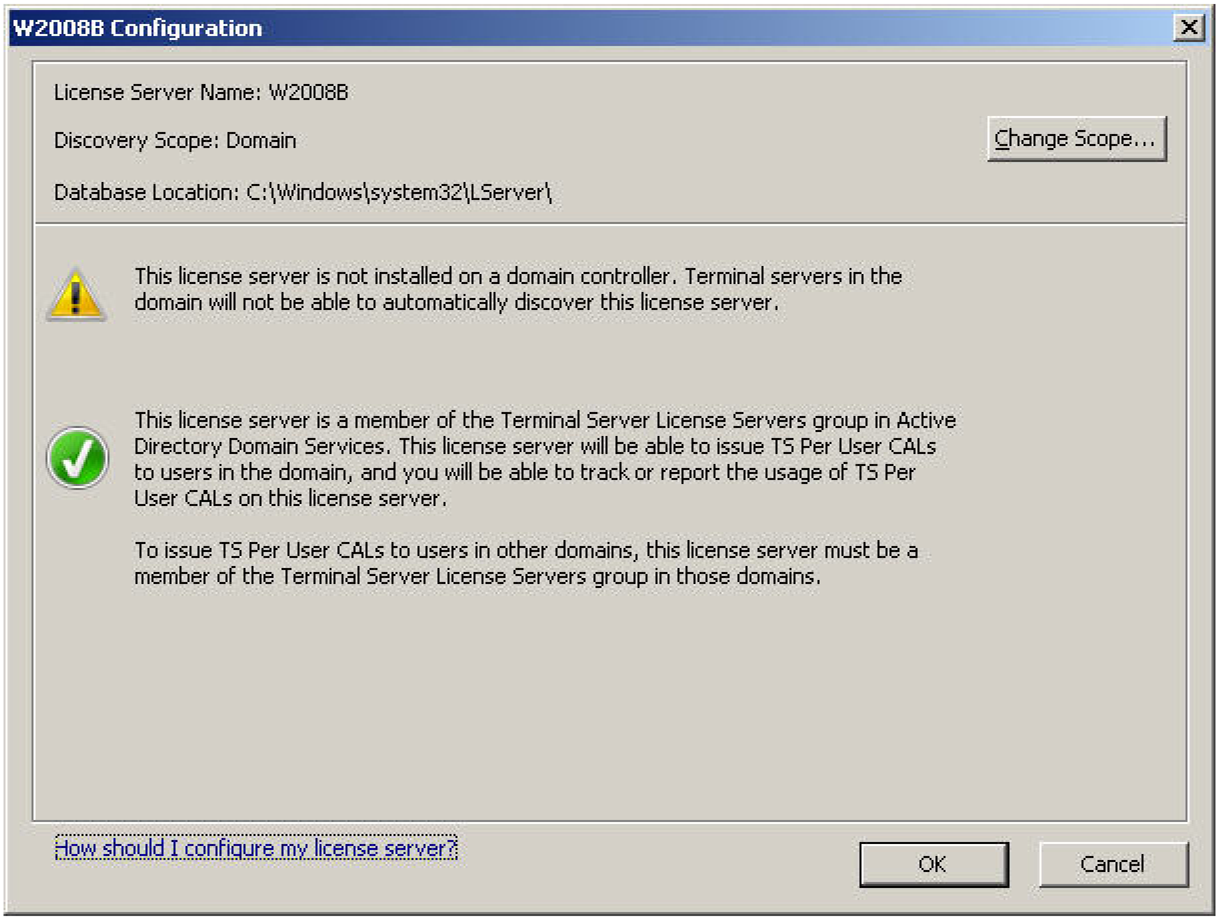
Figure 7.4: An example of the TS Licensing Manager immediately after installation and prior to activation and installation of licenses.

If this server has Internet access, choose *Automatic connection (recommended)* for the *Connection method*. The server will ensure that a connection can be made with the Microsoft Clearinghouse and request Company Information such as name, company, and region as well as optional physical and email contact information. Once entered, the server will register with the Microsoft Clearinghouse and activate the license server.

Step two is to actually install the TS CALs needed by your environment. Once licenses have been purchased either directly through Microsoft or through a partner they will be made available through the Microsoft Clearinghouse. For retail purchased license packs, a license key will be required. For other types of agreements, the agreement number will be required. Once registered, the licenses will be installed to the server.

Licensing for Terminal Services has historically been a confusing process for many administrators due to the idiosyncrasies with the licensing system. Because of these historical problems, two new features with Windows Server 2008 are the *Review Configuration* wizard and the *Licensing Diagnosis* node in Server Manager.

The Review Configuration wizard can be launched from within the TS Licensing Manager console by right-clicking the server name and choosing *Review Configuration*. This tool performs a series of tests against the configuration of the license server itself and notifies the administrator when known issues are seen. Figure 7.5 shows an example of the warning message that appears when the license server is installed with the Domain licensing scope. You'll see there that this is also the location where the licensing scope can be later changed if desired.



License Server Name: W2008B

Discovery Scope: Domain

Database Location: C:\Windows\system32\LServer\

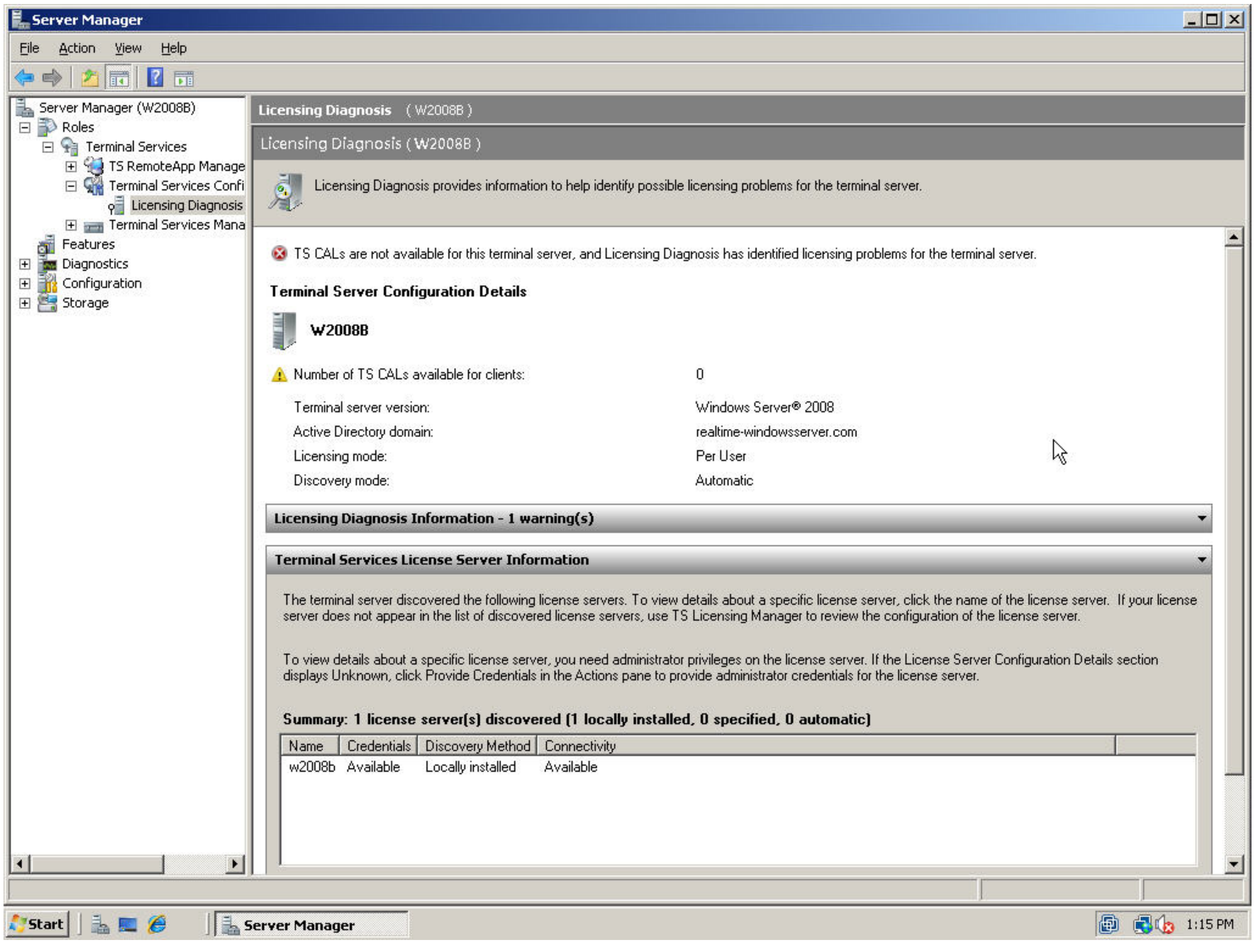
This license server is not installed on a domain controller. Terminal servers in the domain will not be able to automatically discover this license server.

This license server is a member of the Terminal Server License Servers group in Active Directory Domain Services. This license server will be able to issue TS Per User CALs to users in the domain, and you will be able to track or report the usage of TS Per User CALs on this license server.

To issue TS Per User CALs to users in other domains, this license server must be a member of the Terminal Server License Servers group in those domains.

Figure 7.5: The Review Configuration wizard alerts the administrator when the TS Licensing configuration may experience known problems.

Also available is the Licensing Diagnosis node in Server Manager where a more comprehensive view of the licensing configuration is shown. For this server (see Figure 7.6), this screen displays information about the number of TS CALs available for distribution to clients, and any warnings regarding configurations that may impact the ability to serve licenses to clients. Of particularly handy use is the bottom screen where all discovered license servers are displayed. This window is useful for identifying which servers are currently serving TS CALs to clients and can be located by this Terminal Server.



Licensing Diagnosis (W2008B)

Licensing Diagnosis (W2008B)

Licensing Diagnosis provides information to help identify possible licensing problems for the terminal server.

TS CALs are not available for this terminal server, and Licensing Diagnosis has identified licensing problems for the terminal server.

Terminal Server Configuration Details

W2008B

Number of TS CALs available for clients:	0
Terminal server version:	Windows Server® 2008
Active Directory domain:	realtime-windowsserver.com
Licensing mode:	Per User
Discovery mode:	Automatic

Licensing Diagnosis Information - 1 warning(s)

Terminal Services License Server Information

The terminal server discovered the following license servers. To view details about a specific license server, click the name of the license server. If your license server does not appear in the list of discovered license servers, use TS Licensing Manager to review the configuration of the license server.

To view details about a specific license server, you need administrator privileges on the license server. If the License Server Configuration Details section displays Unknown, click Provide Credentials in the Actions pane to provide administrator credentials for the license server.

Summary: 1 license server(s) discovered (1 locally installed, 0 specified, 0 automatic)

Name	Credentials	Discovery Method	Connectivity
w2008b	Available	Locally installed	Available

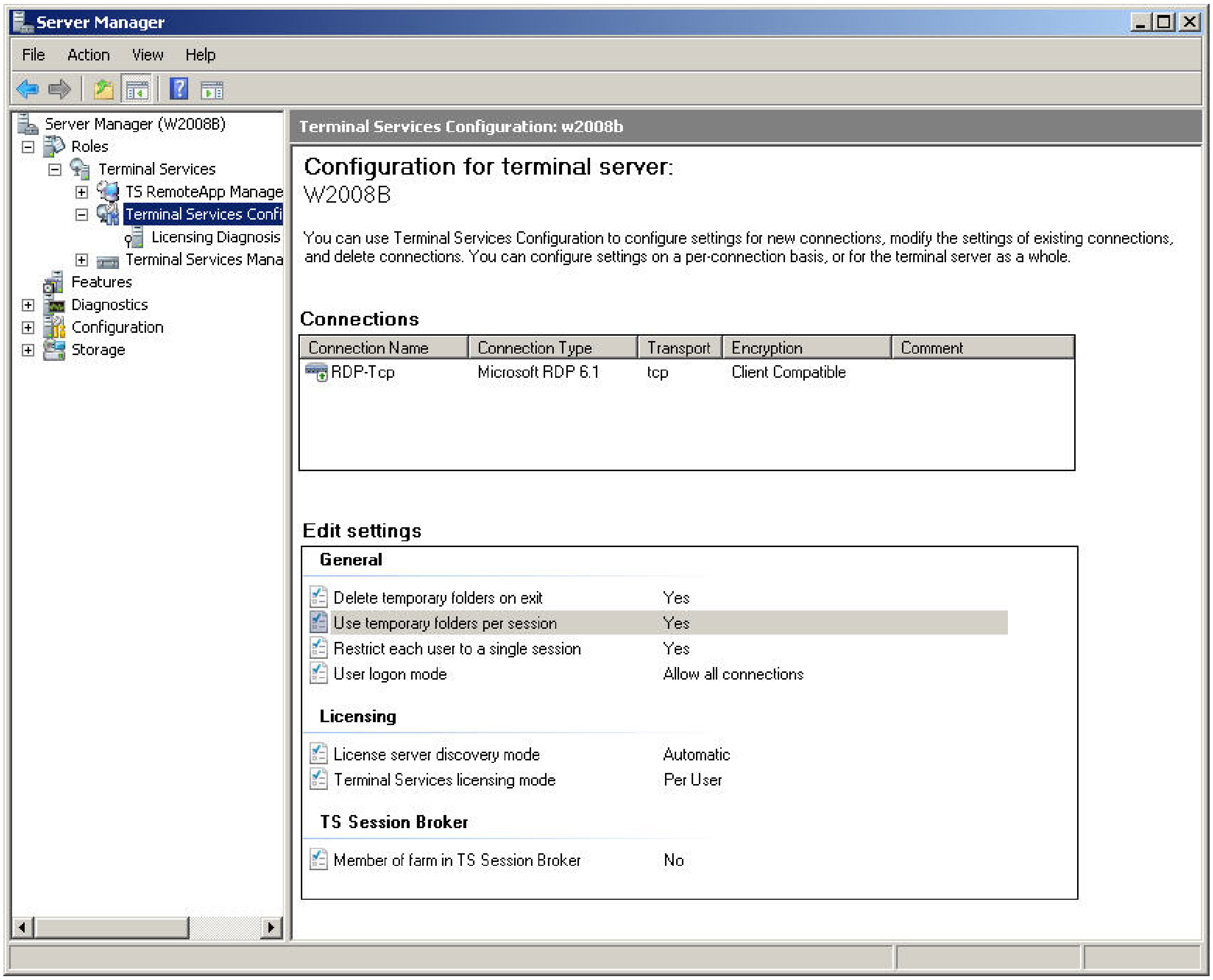
Figure 7.6: The Review Configuration wizard alerts the administrator when the TS Licensing configuration may experience known problems.

Managing Terminal Services

Once Terminal Services has been installed and correctly licensed, there are a number of steps to accomplish to make the server available for use by users. Within Server Manager are a number of configurations that determine how users interact with the server. There are also best practices associated with installing applications and managing user profiles. In this section, let’s discuss each of these management steps in turn.

Server Manager

Upon the installation of the Terminal Server Role Service, Server Manager is extended to include three new nodes: TS RemoteApp Manager, Terminal Services Configuration, and Terminal Services Manager. Figure 7.7 shows an example of this with the Terminal Services Configuration node highlighted.



Configuration for terminal server:
W2008B

You can use Terminal Services Configuration to configure settings for new connections, modify the settings of existing connections, and delete connections. You can configure settings on a per-connection basis, or for the terminal server as a whole.

Connections

Connection Name	Connection Type	Transport	Encryption	Comment
RDP-Tcp	Microsoft RDP 6.1	tcp	Client Compatible	

Edit settings

General

Delete temporary folders on exit	Yes
Use temporary folders per session	Yes
Restrict each user to a single session	Yes
User logon mode	Allow all connections

Licensing

License server discovery mode	Automatic
Terminal Services licensing mode	Per User

TS Session Broker

Member of farm in TS Session Broker	No

Figure 7.7: Initial configuration of Terminal Server is done via Server Manager. In this figure is displayed the Terminal Services Configuration node where RDP protocol and server-specific settings are configured.

Skipping over the TS RemoteApp Manager node until the next chapter, there are a number of configurations of value within Terminal Services Configuration. This console is used to manage the configuration of the RDP protocol itself as well as a few server configurations. If you double-click the *RDP-Tcp* connection within the console, a properties window appears. This window includes eight tabs for configuring the properties of the protocol:

- *General*. This tab provides for configuring the security and encryption level for the protocol as well as identifying the certificate to be used. As we’ll discuss in the next chapter, certain services require the use of a server certificate for authentication and/or encryption. By default a self-signed certificate is available for use; however, a trusted certificate is necessary for production use of these features.
- *Log on Settings*. By default, clients are configured to provide their own logon information. This is set so that each individual client is authenticated based on their own user permissions. Alternatively, in a low-security environment, it is possible to configure the server to automatically logon each client with a preconfigured user account. Doing so eliminates the ability to map individual people to sessions, but enables a type of anonymous logon.

- *Sessions*. By default, settings related to session disconnection, session reset, and idle limits are configured within each individual user's Active Directory object. This tab provides a place to override the user object configuration for all connections to this server. In environments where a cohesive policy is desired for these settings, it is a best practice to enable this override here so that these settings do not need to be configured for each individual user.

> ☞ It is often a best practice at this screen to end disconnected sessions after a short number of minutes (such as 5 minutes) and set an idle limit to a large value (such as 240 minutes). This allows accidentally disconnected sessions to automatically reset after a few minutes, while also resetting sessions that have become idle for a long period of time.
>
> When session limits are reached or connections are broken it is often a best practice to simply end the session here rather than disconnect the session. This is done to free system resources that would otherwise never be released until the session is correctly logged out.

- *Environment.* Three settings are possible in this tab. By default, the configured setting will *Run initial program specified by user profile and Remote Desktop Connection or client*. This setting enables both desktops and TS RemoteApps to be used on this server. Alternate options are to disallow an initial program to be launched, which has the effect of restricting users to full desktops only and to hard code a specific application to be launched at connection in the case where the Terminal Server only hosts a single application.
- *Remote Control*. One of the administrative benefits to Terminal Services is the ability for users and administrators to "look over the shoulder" of another session for troubleshooting or cooperative work. This tab enables the configuration of those Remote Control settings either based on the AD user object, or via an override. As with session information, it is often a best practice to override the user settings at this screen for easier administration.
- *Client settings*. This tab includes the master toggle switches for disabling certain features for all connecting clients including color depth, connected drives, audio, clipboard sharing, and other features.
- *Network adapter*. It is possible in this tab to identify the specific network adapter to use for this instance of the protocol as well as the number of concurrent connections to support on that adapter. The settings here are usually left alone.
- *Security*. Within this tab it is possible to granularly identify which users have what kinds of access to the Terminal Server.

> ☞ One use of this tab is in granting individual non-administrative users the ability to Remote Control other sessions. This is done by granting the Allow Remote Control privilege on the Remote Desktop Users group under the Advanced button.

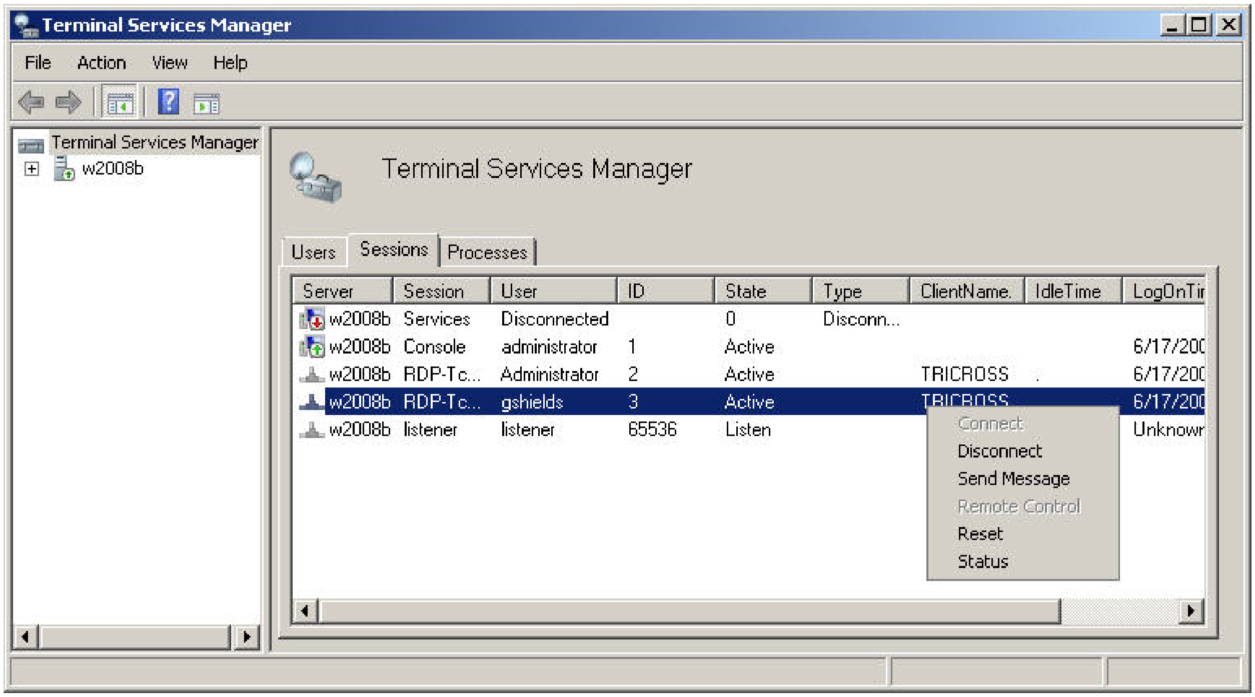
By double-clicking any of the entries in the *Edit settings* section of *Terminal Services Configuration*, this brings forward another properties box. We've already talked about the contents of the Licensing tab, and we'll talk about the TS Session Broker tab in Chapter 8. But the General tab of this box is useful during initial server configuration to control the several settings that can affect the performance of the Terminal Server (and, correspondingly, the experience of its users). Four options are available:

- *Delete temporary folders on exit.* Through the course of daily operations, user sessions tend to accumulate files within their temporary folders. This folder by default is mapped to *C:\Users\{username}\AppData\Local\Temp\1*. The collection of these files over time can fill up the available drive space, which is a particular problem with the default configuration because temporary folders are stored on the system drive. Checking this box instructs the server to empty the user's temporary folders location at logoff.
- *Use temporary folders per session.* The temporary folders path noted in the previous bullet is enabled when this checkbox is selected. This checkbox identifies a separate temporary folder per user, and its selection is a best practice for ensuring that users and their temporary folder usage does not "step on" each other.
- *Restrict each user to a single session.* When this selection is disabled, it is possible for users to open multiple sessions to the same Terminal Server, which can consume an unnecessary excess of resources. There are some situations, however, where multiple sessions may be useful. By default this configuration is selected, but each environment should weigh their need for resource conservation against the restriction against multiple sessions.
- *User logon mode.* This configuration is used during maintenance operations to prevent new logons from occurring to the server. In the case where maintenance is desired, it is not often a best practice to simply kick users off the server if the maintenance activity is of low priority. This setting, new to Windows Server 2008, enables the administrator to prevent any new logons to the server while "draining" existing connections as users naturally complete their work and logoff.

Be aware that the Server Manager version of Terminal Services Configuration discussed earlier and Terminal Services Manager discussed shortly can only work with the local server. By navigating to *Administrative Tools* | *Terminal Services* | *Terminal Services Configuration* or *Terminal Services Manager*, it is possible to manage multiple servers from the same interface. This is done by right-clicking the top-level node and choosing to *Connect to Computer*.

The Terminal Services Manager node is another node available within Server Manager whose primary use is in identifying and interacting with the sessions and users currently logged into Terminal Servers within the domain. Three tabs are available for use by administrators:

- *Users.* Individual users that are currently logged into connected servers are shown in this tab. From this tab, user sessions can be disconnected, reset, or logged off. Administrators can send network messages to specific users from this screen or initiate a Remote Control session.
- *Sessions.* This screen includes all of the currently open sessions on the server and is shown in Figure 7.8. Slightly different than the Users screen, all open sessions including listener sessions and console sessions are shown here. Similar actions can be done to sessions within this screen as can be done on the Users screen.
- *Processes.* Each session on each server is comprised of a number of individual processes. Those processes are what drive the activities being completed by each user. But sometimes those processes use too many resources or need killing due to problems. Within this screen, all processes are listed by their owning user and session. Right-clicking any process allows an administrator to *End Process*.



Server	Session	User	ID	State	Type	ClientName	IdleTime	LogOnTi
w2008b	Services	Disconnected		0	Disconn...			
w2008b	Console	administrator	1	Active				6/17/200
w2008b	RDP-Tc...	Administrator	2	Active		TRICROSS	.	6/17/200
w2008b	RDP-Tc...	gshields	3	Active		TRICROSS		6/17/200
w2008b	listener	listener	65536	Listen				Unknowr

Figure 7.8: A look at the Terminal Services Manager's Sessions screen showing some of the actions that can be done to individual sessions.

Installing Applications

Once the initial configuration of the Terminal Server settings is complete, you are ready to begin installing the applications you wish to host. First and foremost, be aware that some applications do not behave properly when run within a multi-user environment like Terminal Services. Often, these applications inappropriately rely on centralized locations such as the HKEY_LOCAL_MACHINE registry hive to store user-specific information. Some applications like these may require specific tweaking, registry manipulation, or other “hacking” for them to properly function when used within the Terminal Server environment. Testing all applications, and most specifically simultaneous access by multiple users, prior to distribution is critical to ensuring their proper functionality.

That being said, Terminal Services does come equipped with an installation mode that assists with some of these types of incompatibilities. Prior to installing any application to a Terminal Server, you must ensure the following:

- *Ensure that no users are logged into the server*. Prior to installing any application, all RDP sessions to the Terminal Server should be closed and all users logged out other than your user account.
- *Install applications from the console*. It is a good practice to install all applications from the console itself rather than through an RDP session. This is due to how some application installations are coded to work with the console session. With Windows Server 2008, the console session is structured differently than with previous OSs, no longer using what is called “session 0”. Although this change to the structure of the console session reduces the requirement for applications to be installed only via the console, in production it remains a good practice.
- *Enter the server into “install mode”*. Prior to launching the setup file for your application, you must first enter the server into “install mode”. There are two ways to accomplish this. Entering *change user /install* at a command prompt will complete the switch. Alternatively, the Control Panel includes a link called *Install Application on Terminal Server*. This Control Panel option will switch the server into “install mode” and prompt you for the setup file within a wizard format.
- *Install the application.* Complete the installation as necessary.
- *Reboot the computer or return the server to “execute mode”*. Once the installation is complete, the server will need to be returned back to “execute mode”. This process instructs the Terminal Server to stop watching for incoming installations and to process whatever installation it has just logged. Do this by either completing the *Install Application on Terminal Server* wizard or from the command prompt enter *change user /execute*. Any reboot of the server automatically brings that server back on-line in “execute mode”.

☞ This last bullet is an important point because some installations require a mid-installation reboot. The reboot occurs at some point during the installation, and once the server is logged back in the installation continues. The problem with installations that have a mid-install reboot is that the server returns from the reboot back in "execute mode" and not in the proper "install mode". Should your application have an installation of this type, it is critical for you to identify how the installation notifies itself to restart after the reboot. Often, this is done by adding a link to the RunOnce key in the registry, found at HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce.

This "install mode" is necessary because many applications do not have install routines that are considered *Terminal Server-aware*. By default, applications typically install system-wide settings to the HKEY_LOCAL_MACHINE registry hive and user-specific settings into the HKEY_CURRENT_USER hive. For typical servers and desktops that only serve a single user at a time, this behavior is normal and desired. But with Terminal Servers, installing user-specific information to the installer user's HKEY_CURRENT_USER means that other users will not necessarily have the registry information they need to properly run the application.

The switch to "install mode" instructs the Terminal Server to watch for any registry updates to the *HKEY_CURRENT_USER\Software* key. Any updates to that location, typically done during an installation, are then logged to a special Terminal Server key located at *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install*. Within this key, often called the *shadow key*, is stored the registry information needed by users when they later log on.

When the server is returned back to "execute mode" and logons are re-enabled, the Terminal Server will then check each user's HKEY_CURRENT_USER hive as they logon to see if they have the proper information they need. If not, that information is copied from the shadow key location.

✐ As stated in the beginning of this section, some applications simply don't function very well when run atop Terminal Server. The process explained here ensures that most applications function properly, but some remain problematic even with this process and may require further tweaking. Testing of all applications, and specifically testing of multiple, simultaneous user access is critical for all applications.

Managing User Profiles

User profiles and their management are another important part to managing the user's experience with Terminal Services. The reason for this is because user profiles by default are system specific and local to the computer where the user logs in. When a user logs into a Windows Server 2008 computer, that computer creates a profile for them as a subfolder of the location *C:\Users*. This works well in the traditional one-user-at-a-time situation seen on non-Terminal Servers. But can cause issues when used in the Terminal Services environment.

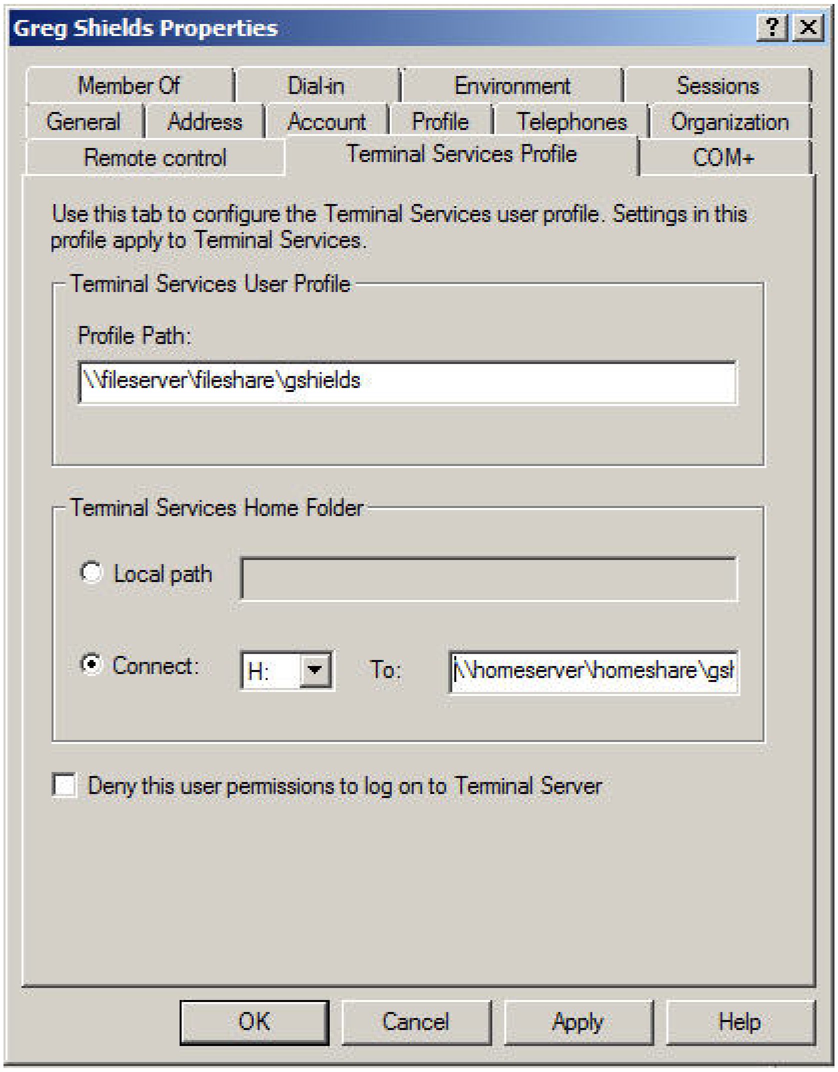
First, local profiles can have a tendency to grow very large. This is particularly the case when users leave large files on their desktop or in profile-housed folders. The relatively few numbers of profiles on typical workstations and non-Terminal Servers usually doesn't cause a problem with disk space usage. But, with large numbers of users potentially logging into a Terminal Server, all of which create a profile, the potential is there for large amounts of disk space to be consumed by user profiles.

> ✎ This has been a known problem with Terminal Servers since inception, and to combat the problem a number of solutions are available. Some solutions utilize third-party integrations to reduce profile size and/or enforce mandatory profiles that never change. Others leverage special coding to merge user specific settings with default or mandatory profiles. All extend the native functionality of Windows to support more efficient use of user profiles.

One solution for assisting with the problems of profiles is to use what are called Terminal Services User Profiles. These special roaming profiles are only used when a user logs into a Terminal Server, and are especially critical when multiple Terminal Servers are available in an environment. Because users expect to see the same environment when they login, Terminal Servers that are load balanced with each other require some form of roaming profile. This ensures that no matter which server the user logs into, they see that same environment.

Terminal Services User Profiles are found within each user's user object in Active Directory Users and Computers (ADUC). Open any user's object within the ADUC console and navigate to the *Terminal Services Profile* tab and you'll see a window similar to Figure 7.9. For each user that will be logging into a Terminal Server, it is possible to identify a file server and file share location which will be the storage location for their Terminal Server-specific roaming profile. As you can see in the figure, it is a best practice to identify a subfolder to the roaming folder share named after the user's username. Also available in this location is a mapping to the user's home drive.

> ☞ Making use of Terminal Services User Profiles is a critical part of Terminal Services administration, most especially in multi-server environments. However, their use will increase the login process as the server copies the roaming profile from the file share. You will need to work with your users to ensure that their profile sizes do not grow excessive, or their login and logoff times can increase substantially.
>
> The careful use of Group Policy and/or third party tools for Terminal Server management is also possible to assist with this process.



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

Terminal Services User Profile

Profile Path: \\fileserver\fileshare\gshields

Terminal Services Home Folder

○ Local path

● Connect: H: To: \\homeserver\homeshare\gsh

□ Deny this user permissions to log on to Terminal Server

Figure 7.9: Terminal Services User Profiles allow users to log into different Terminal Servers and get the same environment.

Using Terminal Services User Profiles enhances the experience for users when they connect to multiple Terminal Servers, but alone this step does not eliminate the problem of profile storage on the Terminal Servers themselves. What is needed along with this step is yet another that instructs the Terminal Server to delete any locally-copied roaming profiles once the user logs off of the system.

This can be done via a Group Policy, found in the Group Policy Management Editor in the location *Computer Configuration | Policies | Administrative Settings | System | User Profiles*. There, enable the policy titled *Delete cached copies of roaming profiles* and attach the Group Policy Object to the Organizational Unit that contains your Terminal Servers. Enabling this policy instructs the targeted machines to delete any locally-copied roaming profile information from *C:\Users* once the user logs off of the system.

> Be aware that this only happens when users log off. If their session is reset instead of going through the logoff process, the server does not complete the deletion. For environments where resets are a regular occurrence, this can impact the processing of profiles.

Printing with Terminal Services

Printing has long been a pain point within Terminal Services environments. Because of the proliferation of printer drivers available on the market, nearly none of which work across multiple types or manufacturers of printers, one historically painful task has been the installation of printer drivers onto the local Terminal Server for use by its users. The reason for this problem is due to how printers are used by local clients. With clients connecting to Terminal Servers from all across the network—and, potentially, the Internet—the use of a print server that is local to the Terminal Server doesn't often connect the user to the printer they want. As an example, if a user in Denver has connected to a Terminal Server in Los Angeles and wants to print a document, they likely want to print it to their local printer and not one that is served off of the print server in Los Angeles.

To enable this to occur, Microsoft long ago enabled the RDC to print jobs to local printers. But those printers could be of multiple manufacturers and/or multiple models, each requiring its own individual driver. Thus, one job of the Terminal Server administrator was to continuously watch the Event Log for error messages that alert for missing printer drivers and subsequently locate, download, and install those drivers. With previous OS versions, this process needed to occur on each individual Terminal Server. The administrator was required to unpack the device driver and from the *Add Printer* wizard install that print driver to the server. Doing this for dozens or hundreds of printers across even a small number of Terminal Servers quickly grew into a management nightmare.

> In fact, such a management nightmare that a number of third party companies have developed solutions to ease the pain of deploying the right printer drivers to Terminal Servers. Those solutions automate much of these manual processes, and are critical in larger environments where printing is necessary.

Another solution for printing problems that arrives with Windows Server 2008 is a new feature called *Easy Print*. This feature adds the ability to eliminate driver installations to Terminal Services completely. Easy Print leverages the XPS print path that is natively available with the RDC v6.1 in Windows Vista Service Pack 1 and is installed to Windows XP with the installation of Service Pack 3. The .NET Framework v3.5 is also required for Windows XP clients.

This print path enables printed jobs to be spooled down to the local client and then be processed by the local client's print driver instead of using a print driver that is installed onto the Terminal Server. Another benefit of Easy Print is that when an RDC client views printer settings, they will see the same printer configuration wizard they are used to seeing on their local machine. This will help eliminate confusion about printer configuration and settings while within an RDP session.

Two Group Policies are available that work with Easy Print. These policies instruct the server how to process printer driver requests from incoming clients. Both are found at the location *Computer Configuration | Administrative Templates | Windows Components | Terminal Services | Terminal Server | Printer Redirection*. The first, titled *Use Terminal Services Easy Print printer driver first*, when enabled instructs the server to attempt to use the Easy Print functionality first before attempting to use any locally-installed printer drivers. This is handy for the situation where clients do not have the proper prerequisite components available for Easy Print functionality such as the right Service Pack or .NET Framework components installed. The second, titled *Redirect only the default client printer*, when enabled instructs the Terminal Server to use only the default client printer. Enabling this second policy is a good practice when possible because of performance issues that can occur during the login process as the server attempts to connect to each client printer.

Summary

With the right administrative configurations in place, the addition of Terminal Services to your Windows Server 2008 infrastructure can significantly benefit the workload of the administrator while extending the reach of corporate applications. As we've seen in this chapter, Terminal Services provides a mechanism for consolidating applications back into the data center not unlike the mainframe days. It reduces the total number of configuration and security touch points required to be managed by IT administrators, while making those applications readily available to users on even slow or latent network connections.

But this explanation of Terminal Services has only begun. Windows Server 2008 adds a number of new and expanded features to Terminal Services that reduces confusion for users, enables seamless connections from client desktops, and provides for security and load-balancing support for high-risk and high-availability environments. In Chapter 8, we'll continue our discussion on Terminal Services and focus on those new and expanded features like TS RemoteApps, TS Web Access, TS Gateway, and TS Session Broker. You'll find that these new additions to the venerable Terminal Server makes its administration more flexible and easier, while providing an enhanced user experience to your users.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.