Chapter 25

Installing, Using, and Administering Remote Desktop Services

Using Remote Desktop Services (RDS)—formerly known as Terminal Services—and an RD Session Host server makes it possible to install and manage applications, or desktops, in one location but be controlled by end users in another location.

Applications that run on the RD Session Host server are called RDS RemoteApp applications. From the end-user perspective, these applications look and feel as though they are running on their local system. The user's keystrokes and mouse movements are sent to the server. Images are sent back to the user's system. Even thin clients can easily run sophisticated applications with ease, though RDS RemoteApps are most commonly run on regular desktop systems.

The old Remote Desktop Services came in two flavors: TS for Administrators and TS in application mode. TS for Administrators is now known as Remote Desktop for Administration, and TS in application mode is known as Remote Desktop Services with an RD Session Host server. Remote Desktop for Administration was covered in Chapter 14, and this chapter covers Remote Desktop Services with an RD Session Host server. Topics include adding the RDS role, configuring an RD session host server, adding RDS RemoteApp applications, and connecting to RDS sessions.

In this chapter, you will learn to:

- Limit the maximum number of connections
- Add an application to an RD Session Host server
- Add a RemoteApp for Web Access
- Add a RemoteApp to the Start menu

Who Needs Remote Desktop Services?

Remote Desktop Services can be used to enable end users to run a Windows-based program on a remote server from their desktop computer. The server hosting the application is called a Remote Desktop Session Host (RD Session Host) server. It's also possible for the end users to access a full desktop session on the RD Session Host server.

As an administrator, you can do the following:

- Deploy and manage applications on a few RDS servers instead of on hundreds or thousands of client computers.
- Provide applications to end users whom you cannot easily support because they're in another office—or another country.

- Reduce the impact of client hardware failures by keeping all applications on a central server. If a client's computer dies, plug in a new one, and they're back to work.
- Avoid misconfigured computers.
- Get out of the hardware rat race that constantly requires more updates to support the latest and greatest software.
- Use computers in environments that are not compatible with desktop computers.
- Simplify help-desk and training support.

If any of these tasks are important to you, then you should seriously consider using Remote Desktop Services with a Session Host server.

Centralized Deployment of Applications

One great benefit to Remote Desktop Services is how it simplifies application deployment. Instead of deploying an application to all the clients using Group Policy or Microsoft System Center Configuration Manager (SCCM), you can install it once on the RD Session Host server.

As an example, your business may have a line-of-business application that 100 users need to access. Instead of installing the application on all 100 desktop computers, an RD Session Host server could be used. The application could be installed once on the server, and each user could then access the application remotely.

Even better, when the application needs to be upgraded or patched, you need to do it only once—on the RDS server.

Supporting Remote Users

Remote Desktop Services can be used for remote access or branch-office access. Some applications have difficulty performing over low-speed connections or need special ports opened on the firewall. Instead of running the application over the low-speed connection, the application can be hosted on the RDS server within a well-connected network.

Clients can still connect via a VPN or low-speed dial-up connection. However, since the application is running on the RD Session Host server in a well-connected network, its performance isn't impacted by the slower connections.

More and more people are telecommuting at least a couple of days a week. Many U.S. government agencies have a legal requirement to support telecommuters, and many telecommuters often don't even have offices or desks. Rather than trying to maintain desktop computers for all the staff, many companies are giving users computers to take home and providing their applications via remote servers.

Supporting PC-Unfriendly Environments

The dream of "a PC on every desktop" will remain a dream, if for no other reason than in some environments the conditions are bad for the desktop PC or the desktop PC is bad for the conditions. In other words, it's not feasible to put a desktop PC anywhere.

Some environments are bad for PCs. PCs don't like dust, excessive heat, or vibration, and you won't like maintaining the PCs if you try to use them in an environment that has any of these characteristics. Of course, PCs can be built to work in extreme conditions such as temperatures as high as 120 degrees or even underwater. And for the companies and people who must have

them in these extreme environments, engineers have engineered solutions—but at a cost. When cost is an issue and a thin client will work, Remote Desktop Services can be a good solution.

We've also seen terminals in health club cafes and coffeehouses set up so that only the monitor is visible, thus reducing the chances of someone dropping a strawberry-banana low-fat smoothie with a shot of wheatgrass juice down the vents. For that matter, if someone does drop the smoothie down the terminal's vents, then, because the applications are installed on and running from the RDS server, replacing the device to provide an identical environment is as simple as unplugging the sticky terminal and plugging in a new one. If you drop a smoothie down a computer's vents, then restoring an identical working environment is significantly more complicated.

What about PCs being bad for the conditions? Clean rooms where chips and boards are made are good candidates for Windows terminals. You can't have dust in a clean room, and the fans in a PC kick up dust. Additionally, becoming sanitized to enter a clean room is neither simple nor inexpensive; you don't want to put devices that need care and feeding from the IT staff in there. Another factor applies to many situations, not just clean rooms: anyplace where space is at a premium is a good candidate for a Windows terminal.

Clients can be running thin clients or just about any desktop operating system including Windows, Linux, and Macintosh (though security is optimized on Windows Vista or Windows 7).

This section isn't to sell you on the idea of Windows terminals but to point out that sometimes they're useful, even required—and you can't use them without an RDS server.

Real World Scenario

Power Struggles

Another aspect of the environment-unfriendly PC applies to the power a desktop PC uses. Several studies have been published on the cost savings of thin clients vs. desktop PCs. With the cost of power these days, the savings can be significant.

One study titled "Power to the People: Comparing Power Usage for PCs and Thin Clients in an Office Network Environment" by Stephen Greenberg, Christa Anderson, and Jennifer Mitchell-Jackson (`www.thinclient.net/power/Power_Study.pdf`) shows some of the possibilities. For example, a single thin client averaged about 10 watts a day while a desktop PC averaged 69 watts. This doesn't include the monitor, but both thin clients and PCs can use low-power LCD monitors instead of the power-hungry CRTs of the past.

The study estimated the cost of power at .10 per kWh and .20 per kWh, which is a good range of power costs within the United States. For 100 clients, this equated to savings of between $3,000 and $6,000 annually.

Saving on power costs isn't the only reason to use Windows terminals, but if you're tossing around the idea of replacing PCs with terminals, it's a compelling argument in favor of it.

Reducing Hardware Refreshes

Does it take a 2.5GHz Pentium with 3GB of RAM installed to check email, do accounting, and poke around on the Web a bit? Of course not, but, as of mid-2009, that's not an unusual hardware profile for a desktop computer. Not that these computers are too expensive in absolute terms;

we're wryly amused that every time we buy a new computer, we pay less for a system more powerful than the last one we bought.

Still, even though they're not too expensive in absolute terms, the new computers aren't always worth it because what you're doing doesn't demand all that much from your hardware. Ironically, unless your job is something demanding such as computer-assisted design, you're often more likely to need a powerful computer at home than at work because game hardware requirements are so high. It takes more computing power to play a few swift rounds of the most recent version of WarCraft than it does to write this chapter. (Fighting orcs is hard work!)

The trouble is, sometimes you do need those more powerful computers if you're planning to keep up with existing software technology. True—you don't need the world's fastest computer to do word processing. You may, however, need a computer faster than the one you have if you're going to keep up with the latest and greatest word processing package that everyone is using. If you want to be able to read all those charts and graphs, you can't always do it when the word processor you're using is six years old, even if it still suits your in-house needs. And you can't always run that new word processor if your computer is six years old.

However, if you're using Remote Desktop Services with an RD Session Host server, the client only displays applications running on the RDS server, rather than running them locally—you don't have to concern yourself with whether the applications will run on the client computer, just the server. If the application will run on the RD server and the client can get to the RD server, then the application will display on the client.

Simplifying the User Interface

Another potential benefit to Remote Desktop Services is it can simplify the user interface (UI). Using a computer isn't as easy for everyone as the marketing world would have you believe. Experienced users find it easy to customize their interface, but those who are less experienced find all sorts of pitfalls when it comes to using their computers: so many options that they get confused and too many ways to break something. Colorful icons with rounded corners do not a simple UI make.

If the people you're supporting need only a single application, then you can save yourself and them a lot of grief by providing a connection that runs this application in a remote desktop and nothing else. This is particularly true with Windows-based terminals, which are little more than a monitor, a box, a keyboard, and a mouse.

Or, if the users are already running a desktop operating system, you can use RemoteApp applications. RemoteApp applications deployed via RDS are as easy to use as any other applications on the end user's computer. RemoteApp applications can be launched from the Start menu, from a desktop icon (of an `.rdp` file), or from a web page.

Providing Help-Desk Support

Finally, Remote Desktop Services can make application support easier, not just in terms of installing new applications and applying fixes but in helping people learn to use those applications. Remote Control lets help-desk personnel or administrators connect to another person's remote session either to watch what they're doing or to interact with the session. (This isn't the security hole it may seem—permissions to do this can be controlled.)

When you have remote control of another user's session, you can either watch what they're doing and coach them (perhaps over the telephone) or actually interact with the session so that

you can demonstrate a process. This beats standing over someone's shoulder saying, "Click the File button at the top left. No, File. The FILE button," or trying to figure out what they're doing when your only information comes from their description of the screen.

Deploying RDS RemoteApp

RemoteApp programs are applications that are running on the RD Session Host server but appear to the end user to be running on their desktop. This is often easier for an end user to conceptualize. They don't have to manage multiple desktops but instead can simply launch another application from their main desktop.

Windows Server 2008 introduced RemoteApp programs, and they've been improved in Windows Server 2008 R2. It does take a little bit of configuration to support RemoteApp programs. Once you've configured all the pieces, users can access RemoteApp applications using the following methods:

Through a web browser If RD Web Access is configured, users can access the web page and click a link to launch the application.

Using a Remote Desktop Protocol (.rdp) file Users can simply double-click a properly configured `.rdp` file to launch the RemoteApp application.

Through the Start menu or a program icon RemoteApp applications can be installed using traditional Windows Installer (`.msi`) packages (also called Microsoft Installer packages). Once installed, users can launch the applications just as any other installed application.

You'll learn how to install all the components and deploy RemoteApp applications for each of these methods in the section "Adding Remote Desktop Services" later in this chapter.

Understanding the Remote Desktop Services Processing Model

Thin-client networking or *server-based computing* (same thing, different emphasis) refers to any computing environment in which most application processing takes place on a server enabled for multiuser access, instead of a client. The terms refer to a network by definition, so that doesn't include stand-alone small computing devices such as personal digital assistants (PDAs) or handheld PCs, although you can add thin-client support to some of these devices.

What makes thin-client networking and computing "thin" is neither the size of the operating system nor the complexity of the apps run on the client, but how processing is distributed. In a thin-client network, most if not all processing takes place on the server. Instructions for creating video output travel from server to client, mouse clicks and keystrokes pass from the client to the server, and all video output is rendered on the client.

Son of Mainframe?

You may have heard thin-client networking described as "a return to the mainframe paradigm." (We have heard this less politely phrased as "You just reinvented the mainframe, stupid!") This comparison is partly apt and partly misleading. It's true that applications are stored and run on a central server, with only output shown at the client.

Netbooks and Thin Clients

Netbooks are exploding on the scene, and you may be wondering how they may fit in here. In case you've just gotten off a deserted island, a netbook is a small (7- to 10-inch screen) portable computer designed for communication on the Internet (hence the *net* in netbook). They have more resources than a thin client but significantly less than a full-blown desktop PC. Because of their size, they are highly mobile. They use less processing power, less RAM, and simpler graphics, which all contribute to using less power and to a longer battery life.

It's entirely possible to use netbooks as part of a Remote Desktop solution. The netbook could connect to the Remote Desktop server either directly over the Internet using RD Gateway or via a VPN. The applications or desktops can be executed on the Remote Desktop server so that the netbook's hardware resources aren't overly taxed.

However, the applications being run in the thin-client environment are different from those run in a mainframe environment; mainframes didn't support word processing or slide show packages, and the video demands on the graphical Windows client are necessarily greater than they were with a text-based green-screen terminal. Yet the degree of control that thin-client networking offers is mainframe-like, and we've heard one person happily describe thin-client networking and the command it gave him over his user base as "a return to the good old mainframe days."

Why the move from centralized computing to personal computers and back again? Business applications drove the development of PCs—the new applications simply couldn't work in a mainframe environment. Not all mainframes were scrapped, by any means, but the newer application designs were too hardware-intensive to work well in a shared computing environment. But those applications came back to a centralized model when it became clear that the mainframe model had some things to offer that a PC-based LAN did not:

- Grouping of computing resources to make sure none are wasted
- Centralized distribution and maintenance of applications
- Clients that don't have to be running the latest and greatest operating system with the latest and greatest hardware to support it
- Client machines that don't require power protection because they're not running any applications locally

All in all, reinventing the mainframe has its advantages. Just as PCs didn't replace mainframes, server-based computing isn't replacing PCs. However, it's nice to have the option to use server-based computing when it makes more sense than installing applications on the desktop.

Anatomy of a Thin-Client Session

A thin-client networking session has three parts:

- The *RDS server*, running a multiuser operating system
- The *display protocol*, which is a data link layer protocol that creates a virtual channel between server and client through which user input and graphical output can flow
- The *client*, which can be running any kind of operating system that supports the terminal client

These are explained in detail in the following sections.

The RDS Server

Remote Desktop Services is one of the optional components you can choose to install on Windows Server 2008 R2. If you've added the Remote Desktop Services role, RDS begins listening at TCP port 3389 for incoming client connection requests as soon as the server boots up and loads the core operating system.

Understanding Sessions

When a client requests a connection to the server and the server accepts the request, the client's unique view of the RDS server is called its *session*. In addition to the remote sessions, a special client session for the console is created.

> **Desktop PCs Can't Run RDS**
>
> Some have asked whether there's any way to make Windows XP, Windows Vista, or Windows 7 into a multiuser server (of sorts). Nope—no Microsoft desktop operating system includes full-fledged Remote Desktop Services, and there is no way to add it. Windows XP, Windows Vista, and Windows 7 all include the Remote Desktop feature that allows someone to connect to the computer via the RDP display protocol. However, only one connection is supported at a time. The Remote Desktop Services feature we discuss in this chapter is solely a server-class feature.

All sessions have unique session IDs that the server uses to distinguish the processes running within different RDS sessions on the same computer. In this context, processes are roughly equivalent to executable files. When a client connects to the RDS server, a session ID is created for the session.

Figure 25.1 shows the Remote Desktop Services Manager monitoring several sessions running on an RDS server.

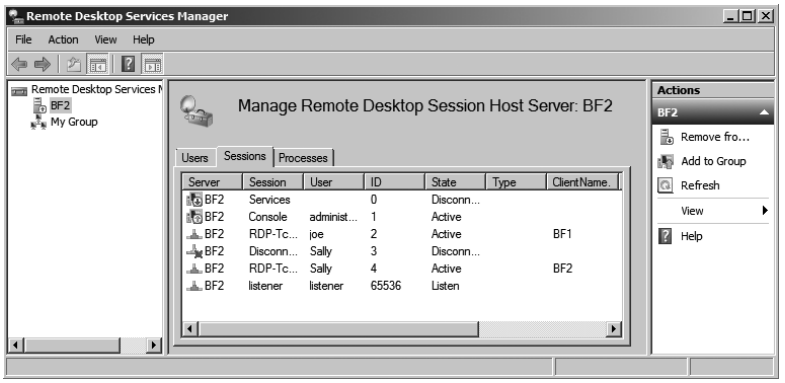


Figure 25.1 Remote Desktop Services Manager

Every desktop session has several base processes running within it to support the user. Additional processes in the session will depend on the applications the user is running.

Executables, Images, and Threads

In Windows operating systems, an executable file is internally known as an *image*. This is because, technically speaking, an application isn't the piece getting processor cycles but instead is a collection of commands called *threads* that get processor time to do whatever they need to do. The threads have an environment called the *process* that tells them where to store and retrieve their data. The part of the process that does something is collectively called the *image* or *executable*. For the sake of consistency with the interface, we'll refer to programs running on the RDS server as *processes*.

The session keeps per-session processes from corrupting each other or viewing each other's data. However, although the sessions are allowed to ignore each other, they still have to coexist. All sessions use the same resources—processor time, memory, and operating system functions—so the operating system must divide the use of these resources among all the sessions. To do so, the RDS server identifies the processes initiated in each session not only by their process ID but by their session ID as well.

Each session has a high-priority thread reserved for keyboard and mouse input and display output, but ordinary applications run at the priority they'd have in a single-user environment. Because all session threads have the same priority, the scheduler processes user input in round-robin format, with each session's input thread having a certain amount of time to process data before control of the processor passes to another user thread. If the sessions are very active, they'll be much more competition for processor time.

The number of sessions an RDS server can support depends on how many sessions the hardware (generally memory but also processor time, network bandwidth, and disk access) can support and how many licenses are available. When a client logs out of her session, the virtual channels to that client machine close, and the resources allocated to that session are released.

The Remote Desktop Protocol

You can run all the sessions you like on the RDS server, but that won't do you any good unless you can view the session output from a remote computer and upload your input to the terminal server for processing. The mechanism that allows you to do both is the *display protocol.*

How RDP Works

A display protocol downloads instructions for rendering graphical images from the terminal server to the client and uploads keyboard and mouse input from the client to the server. Remote Desktop Services natively supports the Remote Desktop Protocol (RDP). RDP provides a point-to-point connection dependent on TCP/IP that displays either the desktop or a single application on the desktop of a client running RDP.

The processing demands placed on the client are reduced by a feature called *client-side caching* that allows the client to "remember" images that have already been downloaded

during the session. With caching, only the changed parts of the screen are downloaded to the client during each refresh. For example, if the Microsoft Word icon has already been downloaded to the client, there's no need for it to be downloaded again as the image of the desktop is updated. The hard disk's cache stores data for a limited amount of time and then eventually discards data using the least recently used (LRU) algorithm. When the cache gets full, it discards the data that has been unused the longest in favor of new data.

Automatic Refresh Timing

The image on the screen is updated at very short intervals when the session is active. If the person logged in to the session stops sending mouse clicks and keystrokes to the server, then the RD server notes the inactivity and reduces the refresh rate until client activity picks up again.

Note that in addition to each client session, there's also a session for the server's use. All locally run services and executables run within the context of this server session.

RDP Version 6.0 and Version 6.1

RDP has been around since NT 4.0 days (the first version was RDP 4.0) and has had a lot of upgrades. The current version available with Windows Server 2008 R2 is version 6.1.

Version 6.0 came out with Windows Vista. The biggest change was the ability to connect to individual applications using RemoteApp instead of launching a full desktop. It also provided support for monitor spanning or using multiple monitors in remote sessions.

Microsoft released version 6.1 in February 2008 and included it with Windows Server 2008. Mostly, this was used to provide support for advanced features to Windows XP SP3 clients. If you're running Windows XP SP3, you should check out the Knowledge Base article at `http://support.microsoft.com/kb/952155`.

Server and Client Requirements

The computing model for thin-client networking means that the horsepower is concentrated on the server end, not the client end. Because the server will be supporting dozens of people—maybe hundreds—this is not the time to skimp on power.

Server Hardware

The notion of using a bigger server so that you can skimp on client-side hardware isn't new. That's all a file server is: a computer running a big, fast hard disk so that you don't have to buy big, fast hard disks for everyone in the office. RDS servers are designed on a similar principle—if most of the processing takes place in a single location, you can concentrate the hardware resources needed to support that processing in a single location and worry less about power on the client end.

Use a Powerful RD Session Host Server

Since an RD Session Host server will be serving applications or full desktops to clients, you'll need to purchase or build a powerful server. Processing power and RAM are the most important resources. Depending on the types and number sessions you're supporting, you may also want to consider boosting disk access and network bandwidth.

On the surface, calculating the needs seems straightforward. Just follow these steps:

1. Calculate the resources needed for the operating system.
2. Calculate the resources needed for a small number of sessions (such as five).
3. Multiply the resources needed for your sessions based on the total number of sessions you plan to support. If you planned to support 100 sessions and you measured five sessions, you'd multiply by 20 (20 * 5 = 100 sessions).
4. Add the total session resources needed for sessions to the resources needed for the operating system.

Although this seems like simple math, it never seems to work out that way. Synergy is often hard to predict. Synergy (where the whole is greater than the sum of its parts) often results in something unexpected. Additionally, if the deployment is successful and users are happy with what they can do, they may end up using it much more than you anticipated.

You don't need to tell this to the budget people, but it's best to add a buffer for the unknowns and to plan for expansion. Additionally, you should do some independent research starting with Microsoft's Remote Desktop Services home: `www.microsoft.com/windowsserver2008/en/us/rds-product-home.aspx`.

Core Hardware Resources

For the purposes of running an efficient RD Session Host server, the bare minimum required to run Server 2008 R2 won't cut it. Although there are no hard-and-fast specifications for an RDS server, some general guidelines for server sizing follow:

Processor Faster is better to a point. More important than a fast processor is one with enough cache so that it doesn't have to reach out to the (slower) system memory for code and data. Faced with a choice between more cache and more speed, go with more cache. Most RDS servers these days have multiple processors, and these processors have multiple cores. Although only multithreaded applications will actually use more than one processor at a time, if there are multiple processors, then threads needing execution can line up at both.

Memory RDS servers tend to be memory bound, not processor bound. Get high-speed, error-correcting memory; get plenty of it; and be prepared to add more as you add more users or applications to the RDS server. The amount of memory you'll need depends on the applications that people use, the number of concurrent sessions, and the memory demands of the files opened in those sessions—computer-aided design (CAD) programs will stress the system more than, say, Notepad. Thankfully, the 64-bit operating system goes well beyond the 4GB limit. Start your calculations with at least 8GB of RAM for the server, and start adding based on the of number of users and memory required by the applications they'll run on the server. Windows Server 2008 R2 will support up to 2TB of RAM.

Disk Consider Serial Computer System Interface (SCSI) disks on an RDS server if at all possible. A SCSI disk controller can multitask among all the devices in the SCSI chain. Most people believe that SCSI performs much better both Serial Advanced Technology Attachment (SATA) and Enhanced Integrated Drive Electronics (EIDE) disks, though some people are starting to find that high-end SATA solutions perform better than low-end SCSI solutions. Disk performance is an important capability in any server, especially so in an RDS server. Additionally, consider a Redundant Array of Inexpensive Disks (RAID) solution to increase the performance and/or fault tolerance of the drives. For a high-end RDS server, a RAID 1+0 solution provides both performance gains and redundancy.

Network On a busy RDS server, consider load-balancing high-speed network cards, which can assign multiple NICs to the same IP address and thus split the load of network traffic. Another alternative is a multihomed server with one NIC dedicated to RDS session traffic. As far as network *speed* goes, sending application output and client-side input back and forth requires little bandwidth, but client-print jobs sent to mapped printers can take quite a bit of bandwidth. Mapped drives may also increase the load by making it possible to copy files back and forth across the RDP connection.

Using the Performance Monitor

The Performance Monitor (discussed in Chapter 17) can help you get an idea of how RDS sessions are stressing the server. Server load should scale closely with the number of people using the server; therefore, as long as you pick a representative group of about five people, you should be able to extrapolate your needs for larger groups. The key objects and counters for measuring general server stress introduced in that chapter will help you size your RDS servers. But a couple of Performance Monitor objects are worth examining to give you detailed information for your RDS server.

Performance Monitor Objects Still Called Terminal Services

Although the name of Terminal Services has changed to Remote Desktop Services in Windows Server 2008 R2, it's still called Terminal Services in Performance Monitor. It might look like a typo, but the two objects are called Terminal Services and Terminal Services Session.

First, the Terminal Services object has counters representing the number of active sessions (sessions where the user has connected to the RD Session Host server and successfully logged on), inactive sessions (where the user is still logged onto the RDS server but has stopped using the session), and the total combined.

Besides simply monitoring activity, you could use this to alert you when the number of active session reaches a certain threshold. Say you wanted to know when a server hosts more than 100 sessions. You could do this with a data collector set.

Chapter 17 discussed data collector sets in more depth, but it's possible to set up a simple user-defined data collector set with an alert. This is done by creating the user-defined data collector set manually (not with a template), selecting Performance Counter Alert, and then setting the threshold for the active sessions. You can then set a task for the alert to notify you with a basic script or log the event to a file.

Although you can get some session-level information from the Remote Desktop Services Manager, a performance object called Terminal Services Session provides quite a bit more data. Use the Remote Desktop Services Manager to find the session you want to monitor—sessions are identified in Performance Monitor by their session numbers, not user login name—and then add counters to monitor that session. Each session object has processor and memory counters that should look familiar to anyone who's used Performance Monitor, but it also has session-specific counters such as the ones in Table 25.1. We haven't included all the counters here, just the ones to show you the kind of information that will be useful when you're calculating the load on the server and looking at the kind of performance the sessions are getting.

TABLE 25.1: Key Terminal Services Session Performance Monitor Counters

COUNTER	DESCRIPTION	SEE ALSO
% Processor Time	Percentage of time that all of the threads in the session used the processor to execute instructions. On multiprocessor machines the maximum value of the counter is 100 percent times the number of processors.	
Total Bytes	Total number of bytes sent to and from this session, including all protocol overhead.	Input Bytes, Output Bytes
Total Compressed Bytes	Total number of bytes after compression. Total Compressed Bytes compared with Total Bytes is the compression ratio.	Total Compression Ratio
Total Protocol Cache Hit Ratio	Total hits in all protocol caches holding Windows objects likely to be reused. Hits in the cache represent objects that did not need to be re-sent, so a higher hit ratio implies more cache reuse and possibly a more responsive session.	Protocol Save Screen Bitmap Cache Hit Ratio, Protocol Glyph Cache Hit Ratio, Protocol Brush Cache Hit Ratio
Working Set	Current number of bytes in the Working Set of this session.	Virtual Bytes, Page Faults/Sec

WAIT ON THE LICENSE SERVER

When experimenting with Remote Desktop sessions to find out how many users you'll be able to support for each session, do not set up a license server; let the RDS server issue its temporary 120-day licenses for this purpose. Although this sounds counterintuitive, using the temporary licenses prevents you from unwittingly assigning per-device licenses to test equipment. See the "Licensing Mode" section for an explanation of how licensing and license allocation works.

Client Hardware

When connecting to an RD Session Host server via a native RDP client, you'll most often use a PC with a Windows operating system loaded, a Windows terminal, or a handheld PC using Windows CE.

> **Native RDP Client**
>
> In this context, a native RDP client means one available from Microsoft and thus implies Windows. Although Microsoft does not support other platforms (except for its OS X Macintosh client, available for download at `www.microsoft.com/mac/products/remote-desktop/default.mspx`), Hobsoft link sells a cross-platform (Windows, Mac, Linux, DOS) Java client at `www.hobsoft.com/products/connect/jwt.jsp`, and there is a free Linux RDP client available at `www.rdesktop.org`.

Windows Terminals

In its narrowest definition, a *Windows terminal* is a network-dependent device running Windows CE that supports one or more display protocols such as RDP or Independent Computing Architecture (ICA), the display protocol used to connect to Presentation Server servers. Many Windows terminals also support some form of terminal emulation.

For this section, think of a Windows terminal as any terminal device designed to connect to a Windows RD Session Host server; it can run any operating system that has an RDP client. A Windows-based terminal (WBT) is such a device that's running a Windows operating system locally—CE or (more rarely) Windows XP/Vista for Embedded Systems—and follows the Microsoft system design requirements for WBTs.

The main thing defining a Windows terminal is its thin hardware profile: because the main job of most Windows terminals is to run a display protocol, they don't need much memory or processing power, and they don't use any storage. A Windows terminal includes a processor; some amount of memory, network, and video support; and input devices such as a keyboard (or equivalent) and mouse (or equivalent). The terminals don't generally have hard disks, CD-ROMs, or DVD players. The operating system is stored in local memory. Beyond those similarities, Windows terminals range physically from a "toaster" form factor to a pad to a small box that can attach to the back of a monitor—or even be part of the monitor itself. Some models of Windows terminals are wireless tablets, intended for people (such as doctors and nurses) who would ordinarily use clipboards and folders to store information.

Although most Windows terminals are entirely dependent on their RDS server, a small set of them can run applications locally. The devices still don't have hard disks; the applications are stored in ROM like the operating system. The types of applications available depend on the terminal's operating system, since locally stored applications must run locally instead of just being displayed. Generally speaking, however, it's more common for Windows terminals to depend on an RDS server for applications.

Windows terminals are most popular in environments where people are using a single application, where supporting PCs would be logistically difficult, or anywhere else that PCs aren't a good fit. However, PCs still outnumber Windows terminals as thin clients. Part of this is because many environments can't depend totally on server-based computing. Companies already have

PCs, and unless they're refreshing the desktop entirely, taking away a powerful PC to replace it with a less-powerful terminal doesn't really make sense.

PC Clients

At this point, people are using more than twice as many PCs as Windows terminals for RDS server client machines. This isn't surprising. First, unless they're starting fresh, people already have the PCs. Even though WBTs are a little less expensive than low-end PCs (not much, though), they're still an added cost. Second, not all applications work well in an RDS server environment. It's often best to run some applications from the RDS server and some locally. Unless you're buying new hardware and don't anticipate any need to run applications locally, you're likely to have to work with PCs for at least some of your terminal clients.

To work with Remote Desktop Services, the PCs must be running a Windows operating system, have the RDP display protocol installed, and have a live network connection using TCP/IP and a valid IP address.

Handheld PCs

We're surprised that handheld PCs (H/PCs) aren't more popular than they are, given how handy they are. They're a terrific substitute for a laptop—inexpensive, lightweight, and thrifty with their power so that you can actually use them during the entire flight instead of having to give up two hours after takeoff. (You can also use one on a plane without worrying that the person in front of you will suddenly recline their seat and crack your laptop's display.) Usually, they run Windows Mobile (previously known as Pocket PC). You can use wired, wireless LAN, or dial-up connections to connect to an RDS server.

What an H/PC looks like depends on who makes it. Some (mine among them) look like a laptop's baby brother. Others fold into a little portfolio shape or are a flat tablet. Some are small pocket-sized deals that are too small to really work on. Some—the ones we prefer—have keyboards; others have only pointers. What all this comes down to is that an H/PC isn't really in a position to replace a desktop PC. Instead, it's usually used in cooperation with a desktop machine with which it's partnered.

Adding Remote Desktop Services

You can add the Remote Desktop Services role to any Windows Server 2008 R2 server using Server Manager. Server Manager includes wizards that allow you to add many roles, and you've probably already used it by now.

When adding the RDS role, you'll be prompted to answer some questions. The following sections will give you the knowledge you need to answer these questions and successfully add the role. Some of the topics related to an RD Session Host server installation include the following:

- Additional role services
- Network Level Authentication
- Licensing mode
- Local Remote Desktop Users group membership
- Adding applications

After the role is installed, you'll need to take some steps to configure the server. This section will guide you through the decision-making process and the steps to add and configure the server.

Remote Desktop Services Not Needed for Administrator Connections

Remote Desktop Services is not needed to connect to a server for administrator connections. Chapter 14 covered remotely connecting to a server using Remote Desktop Connection (RDC) or Remote Desktops. To use these tools, you don't need to install Remote Desktop Services. Instead, you only need to enable Remote Desktop connections on the server.

A significant difference between remotely connecting for administrator purposes and using an RD Session Host server is that licenses aren't needed for administrator connections. Any server can support as many as two remote administrator connections without a license. However, licenses are required for RD Session Host server connections on one-to-one basis. In other words, you'll need a license for every connection.

Required Role Services

Remote Desktop Services is a server role and includes several role services. All of the services aren't required for every installation. You'll need to evaluate what you're trying to accomplish to determine which services to add.

Remote Desktop Session Host The RD Session Host service enables the server to host Windows-based programs or a full Windows desktop. This is a required service for the role.

Remote Desktop Virtualization Host The RD Virtualization Host service is integrated with Hyper-V to allow users to connect to a virtual machine on a server hosting Hyper-V. It can be configured so that users will connect to their own unique virtual machine and allow users to run multiple operating systems simultaneously. This service requires the Hyper-V role service and is needed if you are using the Hyper-V role service.

Remote Desktop Licensing The RD Licensing service manages the client access licenses (RDS CALs) that are needed to connect to an RD Session Host server. It's possible to run Remote Desktop Services without licenses for a limited grace period of 120 days. This allows you to deploy, configure, and test the server.

Remote Desktop Connection Broker The RD Connection Broker service is used for session load balancing and session reconnection in an RD Session Host server farm. It's also required to support RDS RemoteApp applications that allow users to launch applications on the RD Session Host server via Internet Explorer.

If you are using multiple RD Session Host servers, the RD Connection Broker can redirect connections to the servers that are the least busy, which provides load balancing. Additionally, if a user is disconnected, the RD Connection Broker will ensure they are reconnected to the same server where their session is active.

Remote Desktop Gateway The RD Gateway service is used to allow users to connect to RD Session Host servers and remote desktops over the Internet. This service requires additional role services including the Web Server (IIS), Network Policy and Access Services, RPC over HTTP Proxy, and the Remote Server Administration Tools.

Remote Desktop Gateway was covered in much greater depth in Chapter 14, including how to add the required services and enable it.

Remote Desktop Web Access The RD Web Access service allows users to access RemoteApp and Remote Desktop Connection through a web browser. If the clients are running Windows 7, they can access these through the Start menu on their system. This service requires additional supporting role services including Web Server (IIS) and Remote Server Administration Tools. IIS is short for Microsoft's Internet Information Services.

Application Compatibility

If you plan on using the RD Session Host server to host applications for end users, you should install it first before installing the applications. Applications that are installed before adding the RD Session Host role may not work correctly in a multiple user environment.

Although some applications will work in multiuser mode even if they've already been installed, many will not. If you've already installed applications that you want to use with RD Session Host server, you should consider uninstalling the application before adding the Remote Desktop Services role.

Easy Print

A neat new feature available since Windows Server 2008 is called Easy Print. Easy Print ensures that client printers are always installed in remote sessions without requiring printer drivers to be installed on the terminal server.

This might not seem like much, but in the past, you were required to install printer drivers on the terminal server for all the printers used by clients. If you have just 50 clients and they were using 10 different print devices, you'd then need to install printer drivers for all 10 print devices on the server even if they were already installed on the client's systems.

Now you may wondering what you need to do to support Easy Print. Almost nothing. The support is there automatically as long as Remote Desktop Services is installed on Server 2008 R2 (or Terminal Services is installed on Windows Server 2008).

Clients need to be running RDC 6.1 and the Microsoft .NET Framework 3.0 with SP1. RDC 6.1 is backward compatible to XP SP3 with a download, as mentioned earlier. Microsoft .NET Framework 3.0 with SP1 is available for download for XP clients from Microsoft's download site: `www.microsoft.com/downloads`. Search for *Microsoft .NET Framework 3.0 Service Pack 1*.

Single Sign-On

Single sign-on is when users are able to provide their credentials once and these credentials are used for the entire session. As long as these credentials have adequate permissions, the user isn't asked to provide their credentials again. Without single sign-on, users can be queried several times to provide the same username and password.

You can implement single-sign for clients that access the RDS server using Windows XP SP3, Windows Vista, and Windows 7 clients or from Windows Server 2008 or 2008 R2 servers.

Two settings are required in the RDP TCP/IP Connection properties. You'll see graphics of these later, but here are the two settings:

- Ensure that the Security layer is set to either Negotiate or SSL (TLS 1.0) on the General tab of the RDP TCP/IP Connection properties.
- Ensure that "Always prompt for password" is not selected in the "Log on" settings of the RDP TCP/IP Connection properties.

Network Level Authentication

Network Level Authentication (NLA) can be used in Remote Desktop sessions to provide better security. When adding the Remote Desktop Services role, you need to specify whether NLA is required. Your decision is based on the clients the RD Session Host server will support.

NLA ensures that the authentication is completed before a full Remote Desktop connection is established. Without NLA, there is a small window of opportunity for a malicious user or malicious software to attack, even if authentication is unsuccessful.

NLA is available by default in Windows Vista and Windows 7. It relies on the Credential Security Service Provider (CredSSP). If all the clients are running Windows Vista or Windows 7, then you should require Network Level Authentication on the RD Session Host server.

Windows XP doesn't natively support NLA. However, if you upgrade to SP3 and enable CredSSP, you can use NLA. You need to modify the registry to use CredSSP in Windows XP SP3. Check out these two Microsoft Knowledge Base articles for more information:

- KB article 951608, Description of the Credential Security Service Provider (CredSSP) in Windows XP Service Pack 3:
  `http://support.microsoft.com/kb/951608/`
- KB article 951616, Description of the Remote Desktop Connection 6.1 client update for Terminal Services:
  `http://support.microsoft.com/kb/951616`

If your clients are older than Windows XP SP3, they cannot use NLA, and NLA should not be required. The older clients will not be able to connect using NLA.

Licensing Mode

You'll be prompted to select the licensing mode when you add the Remote Desktop Services role. The licensing mode specifies what type of Remote Desktop Services Client Access Licenses (RDS CALs) you'll use. You have three choices:

Configure Later You can postpone your decision and simply select Configure Later. You'll have a grace period of 120 days to configure licensing and select a licensing mode. It's common to choose this option early in the deployment cycle and then configure the RDS CALs once you've worked out the kinks in your RD environment.

Per Device A per-device CAL is issued to a client computer or device. If the licensing mode is set to Per Device and a licensing server has been configured, the licensing server will issue the device a temporary license the first time the device connects. The second time the device connects, the licensing server will attempt to issue it a permanent license.

The licensing server will enforce per-device CALs. In other words, if a per-device CAL doesn't exist for the device and an RDS CAL isn't available to be issued, the connection will be blocked.

You should use per-device CALs if multiple users will use the same device to connect to an RD Session Host server.

Per User A per-user CAL allows a user to connect to an RD Session Host server from any number of devices. Interestingly, the license server doesn't track the per-user CALs. This can make things both easier and more difficult. It's easier to manage on a day-to-day basis because the RD Session host server won't stop users from connecting. However, administrators still have a responsibility to ensure that appropriate CALs have been purchased, which does take some extra administration.

It is possible to configure the maximum connections supported by the server to coincide with the number of purchased CALs. This is done on the Network Adapter page of the RDP-Tcp Properties in the Remote Desktop Session Hosts Configuration console. This isn't exact since users can legitimately connect to more than one session at a time unless you've limited users to only a single connection at a time.

You should use a per-user CAL if users will connect to an RD Session Host server from multiple devices.

A Remote Desktop Services Licensing server needs to be configured to install, issue, and track RDS CALs. Clients won't be able to connect to the RD Session Host server if RDS CALs haven't been purchased and added to the licensing server before the grace period.

Remote Desktop Users Group

Users need to be members of the local Remote Desktop Users group in order to connect to the RS Session Host server. You can add them when you add the role or add them later. The Administrators group is added to the Remote Desktop Users group by default.

Two Remote Desktop Users groups exist: one in the domain and a local group on the RD Session Host server. You need to add users and groups into the *local* group to grant access for them to connect.

It's not uncommon for an administrator to incorrectly add users to the domain Remote Desktop Users group thinking this will grant access to the RD Session Host server. After a little bit of head banging or hair pulling, the little word *local* is noticed. Once users are added to the local group, things work just as advertised.

Adding the Remote Desktop Services Role

You can use the following steps to install Remote Desktop Services. A word of warning, though: you really need to install this on a computer that isn't a domain controller. In our example environment, we're using one server as a DC (named BF1) and another server as the RDS server (named BF2) in a domain named bigfirm.com. If you try install RDS on a DC, you'll receive a couple of warnings and later realize some things just can't work. For example, you'll need to manipulate local groups, but local groups don't exist on a DC.

1. Log onto a member server.
2. If Server Manager doesn't launch automatically, launch it by selecting Start ➢ Administrative Tools ➢ Server Manager.

3. In Server Manager, select Roles, and click the Add Roles link.
4. If the Before You Begin page appears, review the information, and click Next.
5. Select the Remote Desktop Services role. Your display will look similar to Figure 25.2. Click Next.

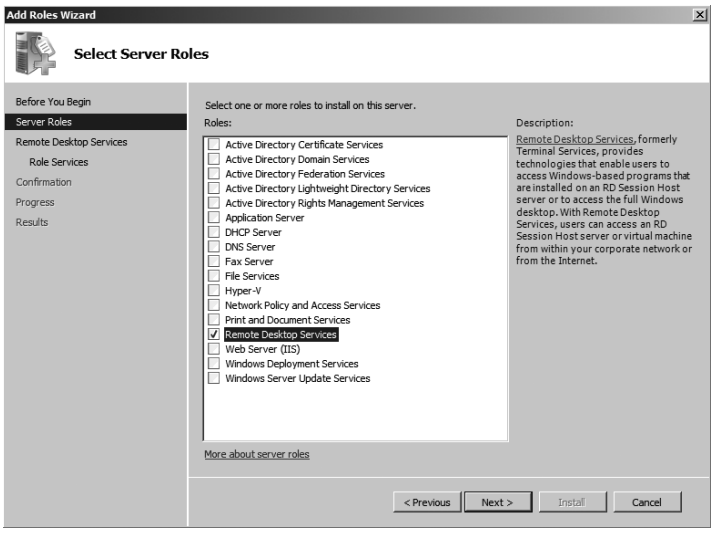


FIGURE 25.2
Adding the Remote Desktop Services role

6. Review the information on the Introduction to Remote Desktop Services page, and click Next.
7. On the Select Role Services page, select the check boxes for Remote Desktop Session Host, Remote Desktop Licensing, Remote Desktop Connection Broker, and Remote Desktop Web Access. When you select the check box for Remote Desktop Web Access, a dialog box will appear similar to Figure 25.3. Click the Add Required Role Services button, and click Next.
8. Review the information on the Applications for Compatibility page. Click Next.
9. Review the information on the Authentication Method page, and select Require Network Level Authentication if your clients are running at least Windows Vista or Windows XP SP3 with the registry modification to enable CredSSP. Select Do Not Require Network Level Authentication if the clients are older. Click Next.
10. On the Specify Licensing Mode page, select Configure Later, and click Next.
11. The User Groups page will appear and includes the Administrators group. You can add users or groups using this page, and they will automatically be added to the local Remote Desktop Users group. Click Next.
12. The Configure Client Experience page will appear, as shown in Figure 25.4. If clients are running Windows 7, you can use this page to enable additional functionality that mimics Windows 7. Select the check boxes for each to install the Desktop Experience feature on the server. It can be disabled later if desired. Click Next.

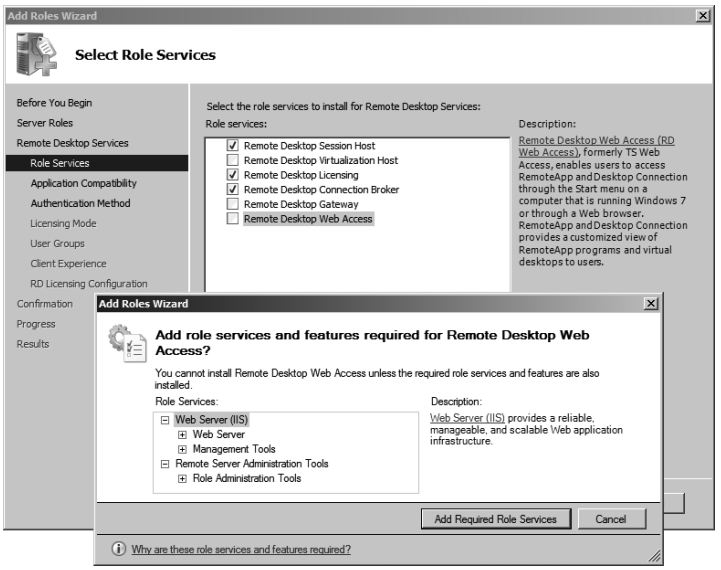


FIGURE 25.3
Adding required role services

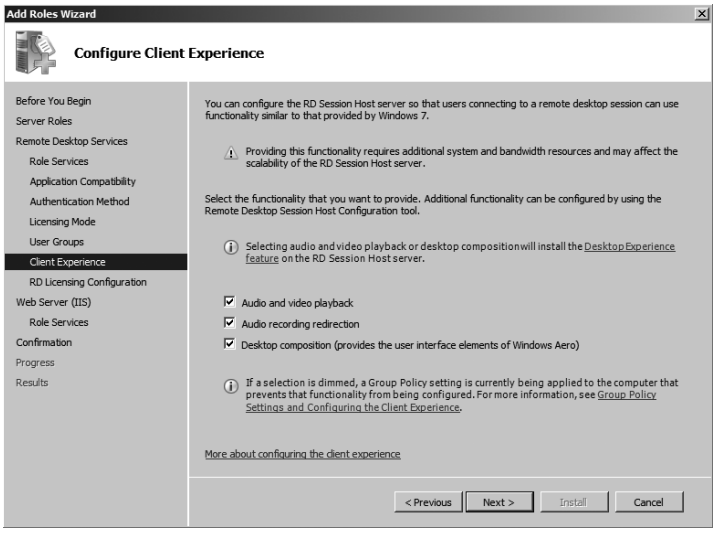


FIGURE 25.4
Configuring the client experience

13. Review the information on the RD Licensing Configuration page. Leave the check box deselected so that discovery scope is not configured for the license server. Click Next.
14. The Web Server (IIS) installation will begin. As a reminder, this is required to support Remote Desktop Web Access. Review the information on the Web Server (IIS) page, and click Next.
15. The Role Services page will show with the required role services already selected. Review the selected services, and click Next.

16. Review the information on the Confirmation page. A warning stating you may need to reinstall existing applications is normal. It's just reiterating that applications installed before Remote Desktop Services is installed may not work in multiuser mode unless they are reinstalled. Click Install.
17. The installation will take several minutes to complete. When it completes, the Installation page will appear indicating a restart is pending. Click Close. When prompted to restart the server, click Yes. The installation will continue during the restart process. This may take several minutes to complete, so feel free to take a cappuccino break.
18. After the system reboots, log on using the same account, and the configuration wizard will resume. When the installation completes, review the information in the Installation Results page, and click Close.

It's normal to get informational messages related to the Remote Desktop Services Server License server since it has not been configured yet. Additionally, you'll see a warning indicating that RD Web Access requires additional configuration, which is covered later in this chapter.

Adding Applications

Although many applications will work automatically in multiuser mode (such as Paint, Calculator, and Notepad), other applications need to be installed. Previous versions of Remote Desktop Services (called Terminal Services) required extra steps to install the applications, but the process is much simpler with RDS.

After the role has been added, you can install any application using an `.msi` (Windows Installer) file or via the Control Panel's Add Remove Programs Wizard. If the application will install via one of these methods, that's all that's necessary.

However, if you have a legacy application that won't install via one of these methods, you'll need to use the `Change User` command. The three-step process is as follows:

1. Execute the `Change User /install` command from the command prompt. This puts the RDS server into installation mode.
2. Install the application.
3. Execute the `Change User /execute` command from the command prompt. This returns the RDS server to the normal mode of operation.

Connecting to an RDS Session

After adding the RDS role, clients that are members of the local Remote Desktop Users group can access desktop sessions on the RD Session Host server. During the installation, you had an opportunity to add users and groups to the Remote Desktop Users group, and you may have done so. Members of the administrators are automatically in the group and can't be removed.

Use the following steps to add a user or group into the local Remote Desktop Users group and verify the RDS server supports remote sessions:

1. Log onto the RD Session Host server.
2. If Server Manager doesn't launch automatically, launch it by selecting Start ➢ Administrative Tools ➢ Server Manager.

3. Browse to the Configuration\Local Users and Groups\Groups node.

4. Double-click the Remote Desktop Users group. Add a user or group from the domain that you'll use to test connectivity. In Bigfirm.com, two users (Joe and Sally) are added, as shown in Figure 25.5. Add any users on your system that you want to be able to access the RD Session Host server. Click OK.

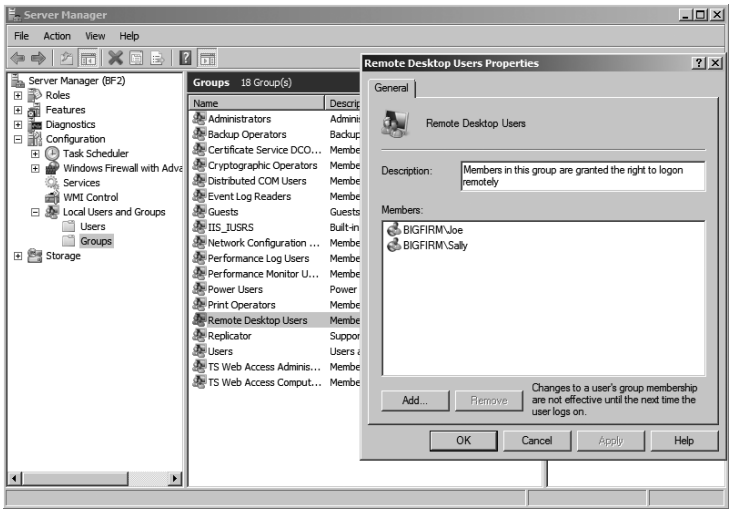


FIGURE 25.5
Adding users to the local Remote Desktop Users group

5. Click Start, type **MSTSC** in the Search box, and press Enter. This will launch the Remote Desktop Connection console. For a full description of all the options available in this tool, you can check out Chapter 14.

6. Click the Options button. Enter the name of the computer hosting RDS in the Computer box, and enter the username of a user in the local Remote Desktop Users group. Your display will look similar to Figure 25.6. Click Connect.

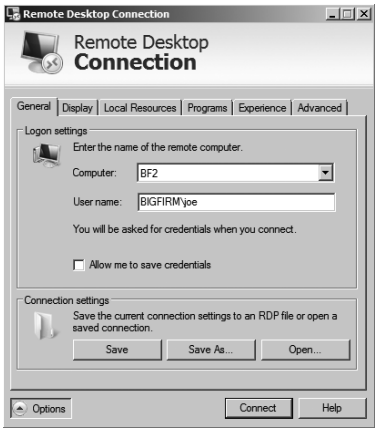


FIGURE 25.6
Connecting with Remote Desktop Connection

7. A Windows Security screen will appear, and you will be prompted to enter the password for the user. Enter the password, and click OK. Security will be negotiated, and after a moment, you will be connected to the desktop.

8. Depending on how MSTSC was configured, the connection may start as a window on the desktop or in full-screen mode. Figure 25.7 shows the connection. Notice it has a Windows 7 look and feel.

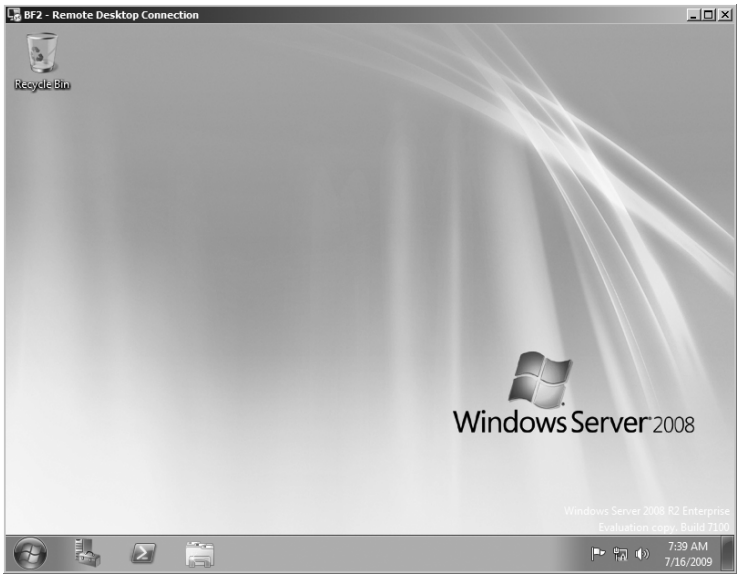


FIGURE 25.7 Connected to the RD Session Host server using Remote Desktop Connection

Although this verifies that you have successfully installed RDS and can connect to RDS sessions, the sessions are still plain desktops. You can install applications on the RD Session Host server and make them available to all users, or you can use RD RemoteApp applications to allow users to run the applications in Windows on their desktops.

Adding an RDS RemoteApp Application

Remote Desktop RemoteApp applications are a neat feature with RDS. Once added and configured, they will run in their own window on the end user's computer. Instead of a user launching a full desktop of another operating system, the RemoteApp application appears just like another application.

Another neat feature is that you can restrict access to the RemoteApp programs by identifying which users and groups can access the program. By default, all authenticated domain users will have access.

There are a lot of steps to get RemoteApp applications to work, but hang in there. None of the steps are that difficult by themselves. As an overview, the following steps need to be taken and are included in this section:

1. Add an RDS RemoteApp program to the RDS server.

2. Add the RDS server to the TS Web Access Computers group.

3. Configure your RD Session Host server to serve RD RemoteApp applications.
4. Identify your RDS server as an RD RemoteApp source.

You'll then be able to launch RD Remote Applications using Internet Explorer from any system in your network.

Adding RemoteApp Programs

Use the following steps to add one or more programs to the RemoteApp program list:

1. Launch the RemoteApp Manager by selecting Start ➢ Administrative Tools ➢ Remote Desktop Services ➢ RemoteApp Manager.
2. The Actions pane on the right includes a link to Add RemoteApp Programs. Click this link.
3. Review the information on the Welcome page, and click Next.
4. The RemoteApp Wizard displays a list of programs that are currently installed on the server and can be added to the RemoteApp Programs list. Select the check box for Paint and any other programs you may want to add. You can also click Browse, and browse to other executable programs on your system that don't show in this list.
5. Click Properties. Select the User Assignment tab. Your display will look similar to Figure 25.8. Notice you can restrict access to programs to specific users and groups here.

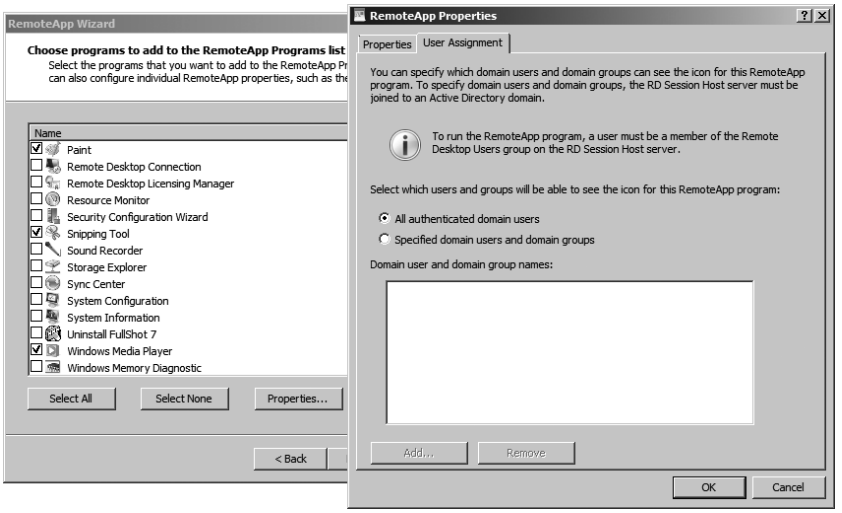


FIGURE 25.8 Adding programs to the RemoteApp Programs list

6. Click OK to dismiss the RemoteApp properties sheet. Click Next on the program selection list page.
7. Review your choices on the Review Settings page, and click Finish.

Adding an RDS Server to the TS Web Access Computers Group

These next steps will add your RDS server to the TS Web Access Computers group:

1. Launch Server Manager by selecting Start ➢ Administrative Tools ➢ Server Manager.
2. Browse to the Configuration\Local Users and Groups\Groups node.
3. Double-click the TS Web Access Computers group, and click Add.
4. In the Active Directory search page, click Object Types.
5. Select the check box next to Computers to include computers in the search. Click OK.
6. Type in the name of the computer hosting RDS, and click OK. Your display should look similar to Figure 25.9. Click OK.

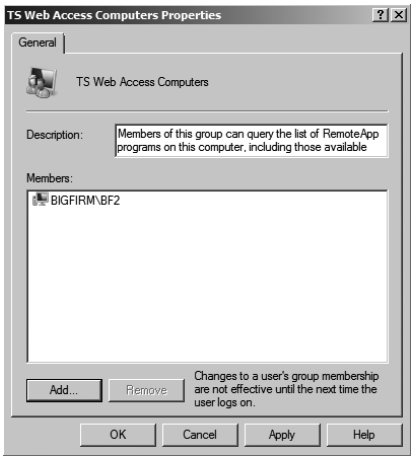


Figure 25.9
Adding the computer to the TS Web Access Computers group

Configuring the RDS Server to serve RD RemoteApp Applications

With the RDS server added to the TS Web Access Computers group, you can now configure the RD Session Host server to serve the RD RemoteApp applications via a web browser.

1. Launch the Remote Desktop Web Access Configuration console by selecting Start ➢ Administrative Tools ➢ Remote Desktop Services ➢ Remote Desktop Web Access Configuration. This will launch Internet Explorer with the address of the RD web server.
2. Unless you've added a certificate from a trusted authority, you will receive an error indicating there is a problem with the website's security certificate. This is normal. The certificate is self-signed, which is good enough for a test environment, but you'll want to install a certificate from a trusted certificate authority for a production server. Click "Continue to this website (not recommended)."
3. After a moment, the Remote Desktop Services Default Connection page will appear. Enter the domain and username for an administrator account and the associated password. Your display will look similar to Figure 25.10. Click the Sign In button.

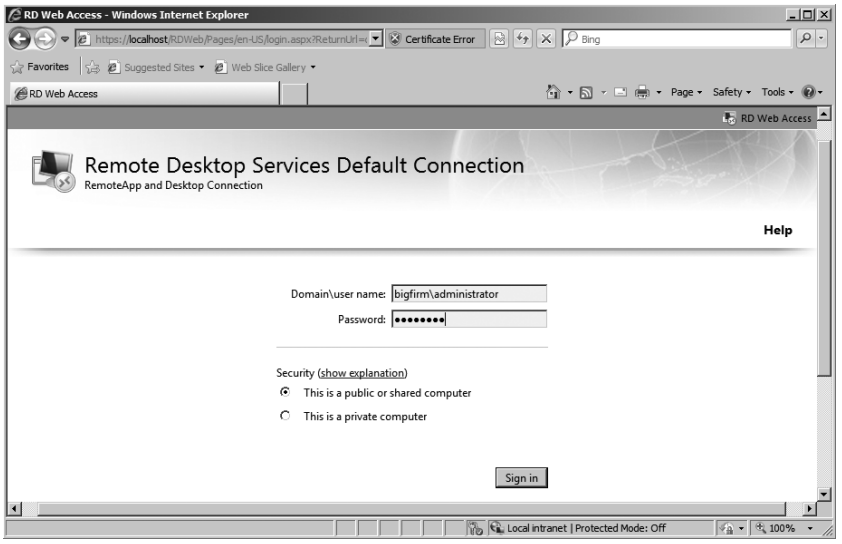


FIGURE 25.10 Remote Desktop Services Default Connection page

4. If you followed the steps in this chapter to install RDS, you included the Remote Desktop Connection Broker as one of the role services installed on the server. This will be the source for your RemoteApp programs. Select An RD Connection Broker Server, as shown in Figure 25.11. Click OK.

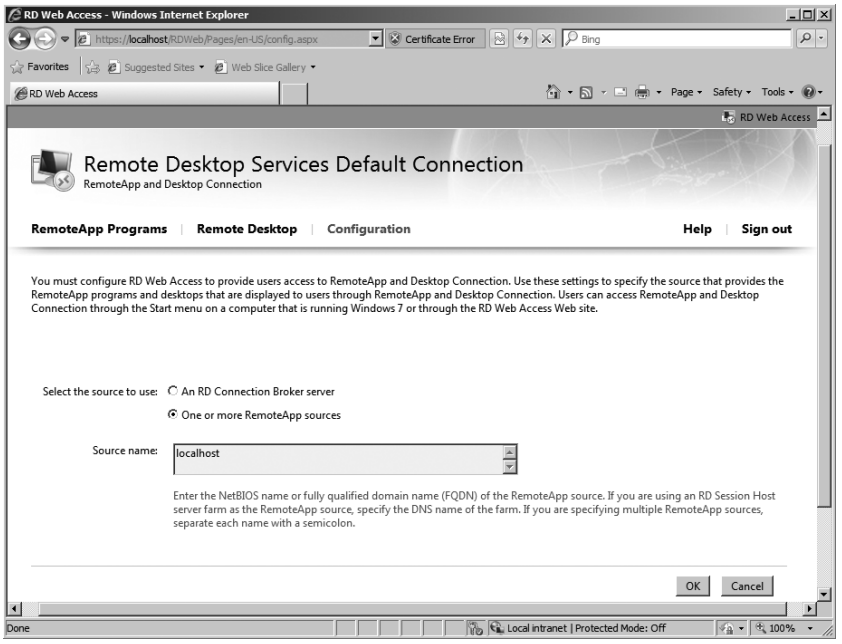


FIGURE 25.11 Configuring RD Connection Broker server as the source

5. At this point, Remote Desktop Web Access is configured, and the Enterprise Remote Access web page will appear with the RemoteApp Programs selected. However, the list is empty.

Even though you added several RemoteApp programs previously, none of them appears because the server hasn't been identified as a RemoteApp source. It's time to do that now.

Adding an RDS Server as a RemoteApp Source

You'll now add your RDS server as a RemoteApp source:

1. Launch the Remote Desktop Connection Manager by selecting Start ➢ Administrative Tools ➢ Remote Desktop Services ➢ Remote Desktop Connection Manager.
2. Select RemoteApp Sources in the navigation tree pane on the left, and then click Add RemoteApp Source in the Actions pane on the right.
3. Enter the name of the server where you've installed RDS. Your display should look similar to Figure 25.12. Click Add.

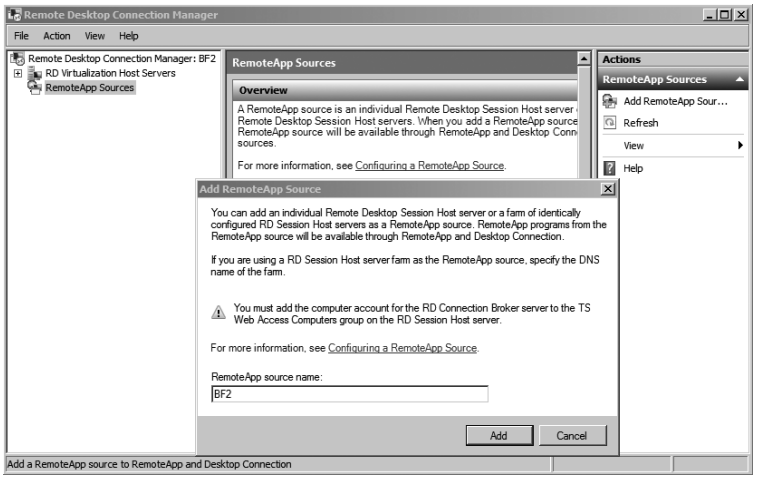


FIGURE 25.12 Using Remote Desktop Connection Manager to add a RemoteApp source

4. Your server will appear as one of the RemoteApp sources. Close the Remote Desktop Connection Manager.

Launching a RemoteApp from Internet Explorer

Launch a RemoteApp from Internet Explorer with the following steps. You can do this from your RDS server, or if desired, you can do it from another computer in your network.

1. Launch Internet Explorer.
2. Enter the following URL into the address bar: **https://localhost/rdweb**.

If you're accessing this from a remote host, enter the name of the server in place of *localhost*. For example, our server name is BF2, so we would enter it as **https://bf2/rdweb**.

3. Since the server is using a self-signed certificate, you'll see an error. Click the Continue to This Website (Not Recommended) link.
4. If prompted by the Internet Explorer Enhanced Security Configuration, click Add to indicate you trust this website. Click Add again, and click Close.
5. The RemoteApp and Desktop Connection page will appear.
6. Enter a username in the format of *domain\user name* and a password for an account that is in the local Remote Desktop Users group of the RDS server. We've created an account named Sally in the Bigfirm.com domain, so we have entered it as **bigfirm\Sally**, as shown in Figure 25.13.

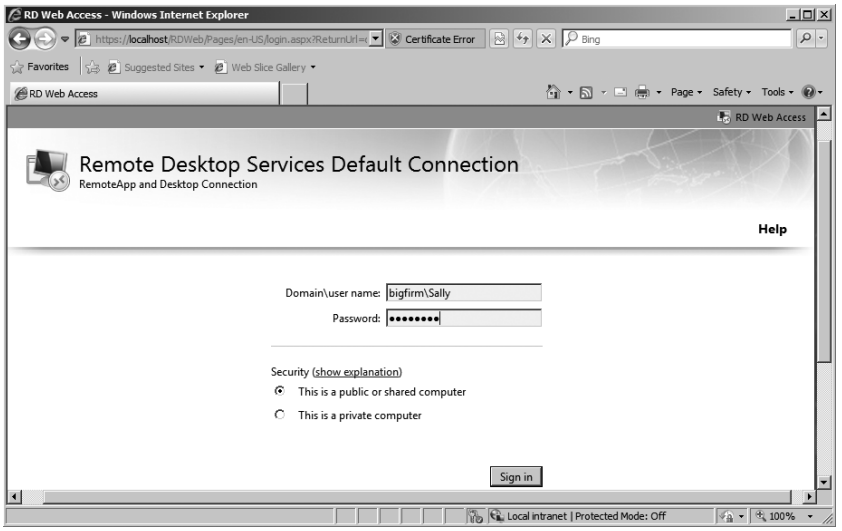


FIGURE 25.13 Logging into the Enterprise Remote Access website

Notice that you can also select whether you're accessing the RemoteApps from a public or private computer. The private setting allows a longer period of activity before logging you off. It's strongly recommended that users close the session as soon as they are finished to flush any remnant data from the session.

7. Enter the user's password, and click Sign In.
8. The RemoteApp programs that have been published to the server are listed, as shown in Figure 25.14.
9. Click the Paint application. A warning will appear providing a warning to the user that the RemoteApp program is starting. Click Connect.
10. Enter the credentials of the same account you used to access the website, and click OK. After a moment, the credentials will be validated, and the Paint program will start.

If you launch Paint from the Accessories menu, you'll see there is very little difference between the local Paint program and the RemoteApp Paint program. They function the same, but the border of the RemoteApp program may be a little different.

FIGURE 25.14 Accessing RemoteApp programs using Internet Explorer

11. Leave the Paint program open, and click another RemoteApp program on the web page. You'll receive the warning again, but after you click Connect, this program will launch without requiring you to enter credentials again.
12. Return to the Internet Explorer web page showing the Enterprise Remote Access menu. Click Remote Desktop.
13. On the Remote Desktop page, click Options. Your display will look similar to Figure 25.15.

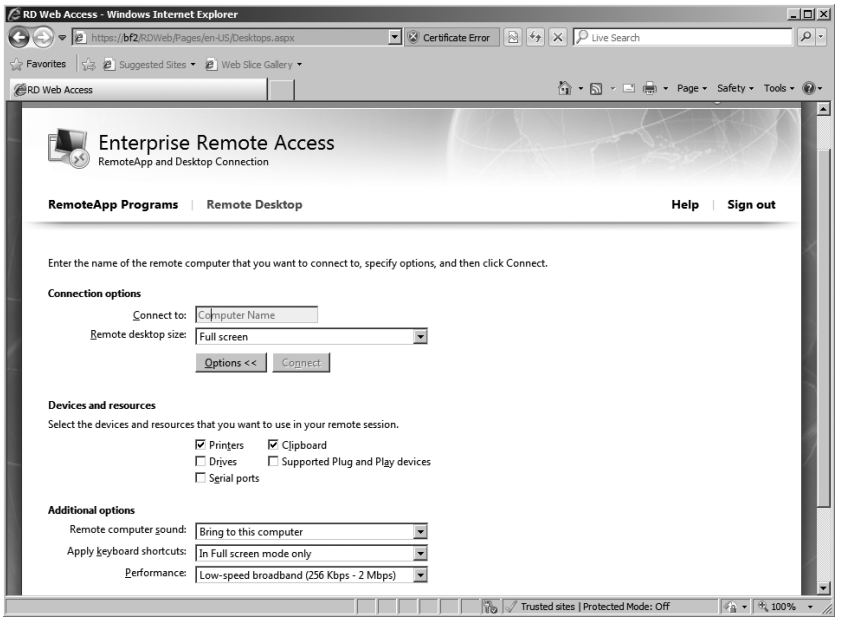


FIGURE 25.15 Accessing Remote Desktop options using Internet Explorer

Notice that you can select many of the same options that are available in the Remote Desktop Connection tool from this page.

14. Enter the name of the RD Session Host server in the Connect To box, and click Connect.
15. A warning will appear as it did before. Review the information, and click Connect.
16. After a moment you will be connected to a full desktop session running on the server.
17. Log off the RemoteApp desktop session, and close all the RemoteApp applications.

Creating .rdp Files for RemoteApp Programs

You can use a Remote Desktop Protocol (`.rdp`) file to allow users to easily connect to an RD RemoteApp application. You can create the `.rdp` file with these steps:

1. Launch the RemoteApp Manager by selecting Start ➢ Administrative Tools ➢ Remote Desktop Services ➢ RemoteApp Manager.

At the bottom of the RemoteApp Manager, you should see one or more RemoteApp programs that were added in previous steps in this chapter. You will also see a warning icon in the Digital Signature Settings area. It indicates a digital certificate has not been configured.

2. Click Change next to Digital Signature Settings to add a digital certificate. Select the "Sign with a digital certificate" check box, and click the Change button. Your display will look similar to Figure 25.16.

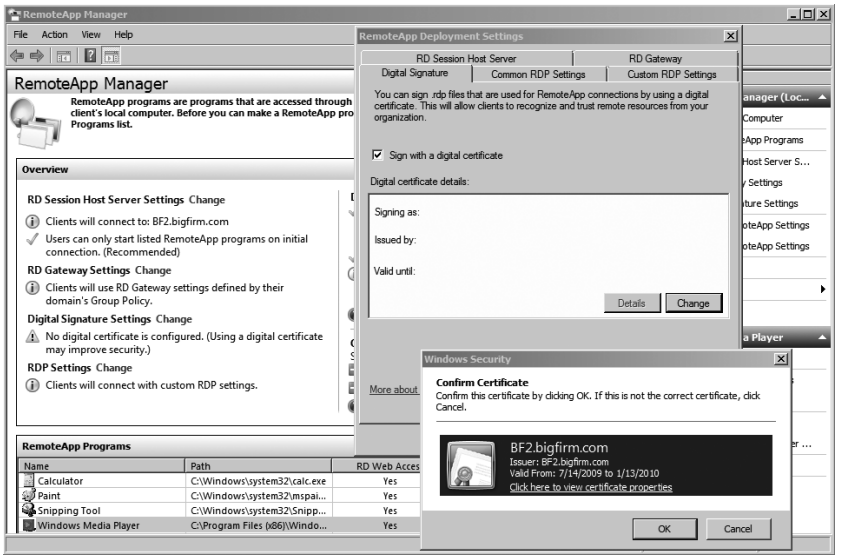


Figure 25.16 Adding a certificate to RemoteApp Manager

Adding the certificate will allow you to sign `.rdp` files, which provides clients an added layer of security.

3. Click OK to confirm the certificate. Click OK to close the RemoteApp Deployment Settings property page. You'll see that the warning icon on the Digital Signature Settings page will disappear.

4. Locate the Paint program in the RemoteApp Program list. Right-click it, and select Create .rdp File.
5. Review the information on the Welcome page, and click Next.
6. The Specify Package Settings page will appear, as shown in Figure 25.17. You can change any of these settings, but the defaults will work for most deployments. Click Next.

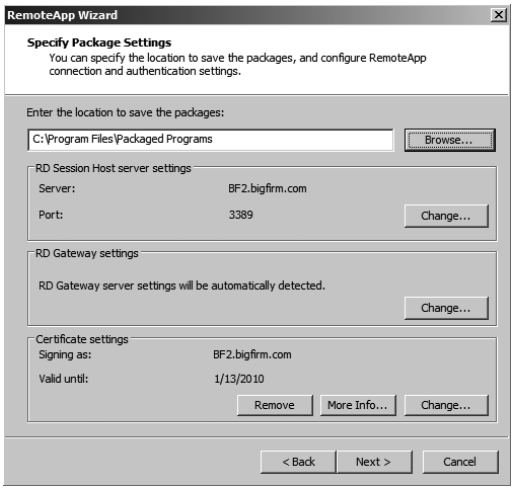


FIGURE 25.17 Specifying package settings for the `.rdp` file

7. Click Finish on the Review Settings page.
8. Windows Explorer will open in the `C:\Program Files\Packaged Programs` folder. It will include the `mspaint.rdp` file.

This file can be copied to other computers or shared. Once it is available to other computers in the network, it can simply be double-clicked to start the application.

LAUNCH .RDP FILES REMOTELY

RDS doesn't seem to like it when you launch RemoteApp programs on the RemoteApp server. If you try to double-click the `.rdp` file on the RDS server, it may work, but often it just doesn't respond. It's best to test your `.rdp` files from another system within the network, even if that other system is a remote desktop launched from within your RDS server.

9. Copy the `mspaint.rdp` file to another computer in your network.
10. Double-click the `.rdp` file on the other computer. Even though the `.rdp` file is signed with a certificate from the RD Session Host server, the server's certificate isn't in the trusted root authority store, so you will receive an error similar to Figure 25.18.

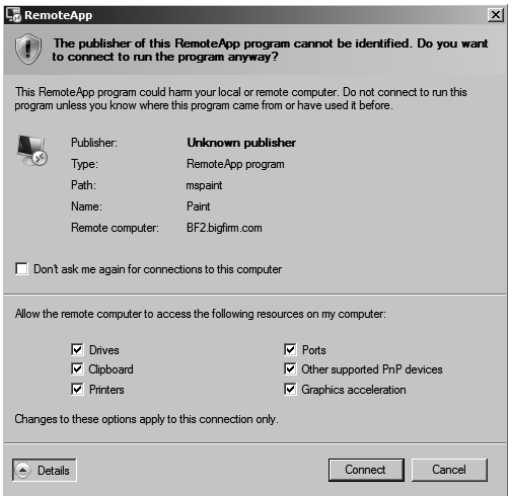


FIGURE 25.18
Unknown RemoteApp publisher warning

If you click the Details page, you can view additional options showing what local resources will be available to the RemoteApp program. Click Connect.

11. Enter the credentials of an account that is in the local Remote Desktop Users group on the RDS server, and click OK. After a moment, the credentials will be verified, the connection will be established, and the program will launch and appear on your desktop.

At this point, you've seen how to launch a RemoteApp application using WebAccess and using an `.rdp` file. Once the program is launched, there isn't any difference in how it works between the two methods.

CREATING WINDOWS INSTALLER PACKAGES FOR REMOTEAPP PROGRAMS

Another way you can deploy RemoteApp applications is by creating a Windows Installer (`.msi`) file and deploying the application using the `.msi` file.

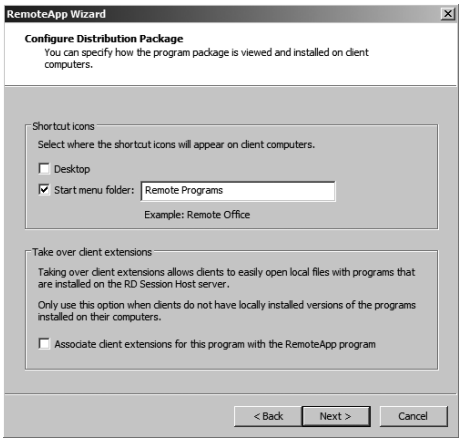
The big benefit of using Windows Installer files is that they can easily be deployed using Group Policy. Once the installer file has been created, you can create GPOs to assign or publish them to users and computers in your domain.

Applications installed with the Windows Installer files can be available via the Start menu and via icons placed on the desktop, depending on what you choose. Use the following steps to create a Windows Installer package for a RemoteApp application:

1. Launch the RemoteApp Manager by selecting Start ➢ Administrative Tools ➢ Remote Desktop Services ➢ RemoteApp Manager.
2. Locate the Paint program in the RemoteApp Programs list. The Paint program was added in previous steps within this chapter. Right-click Paint, and select Create Windows Installer Package.
3. Review the information on the Welcome screen, and click Next.
4. The Specify Package Settings page will appear. This page is the same page that you saw when you created an `.rdp` file. Click Next.

5. The Configure Distribution Package page will appear, as shown in Figure 25.19. Notice that you can add shortcut icons for the program on the desktop and on the Start menu. Additionally, you can have the RemoteApp take over the client extensions. For example, if the `.bmp` client extension was set to launch the local Paint program, you can change it to launch the RemoteApp Paint program instead. Select the "Start menu folder" check box only, and click Next.

FIGURE 25.19
Configuring the distribution package



6. Review your settings, and click Finish.
7. Windows Explorer will open in the `C:\Program Files\Packaged Programs` folder. It will include the `mspaint.msi` file. Additionally, the `mspaint.rdp` file may be in the same directory from the previous set of steps.

You could deploy this Windows Installer file via Group Policy or by simply executing it on a computer.

This file can also be copied to other computers or shared. Once it is available to other computers in the network, it can simply be double-clicked to start the installation.

8. Copy the `mspaint.msi` file to another computer in your network.
9. Double-click the `.msi` file on the other computer. You will receive an error similar to Figure 25.20, but since you created the `.msi` file, you know it is safe. Click Run.

FIGURE 25.20
Windows Installer security warning

10. The Windows Installer file will run and create a shortcut on the start menu. Select Start ➢ Remote Programs ➢ Paint.
11. After a moment, the RemoteApp warning will appear because the certificate is not trusted. This is the same issue you saw with the `.rdp` file in the previous exercise. Click Connect.
12. Enter the credentials of an account that is in the local Remote Desktop Users group on the RDS server, and click OK. After a moment, the credentials will be verified, the connection will be established, and the program will launch and appear on your desktop.

At this point, you've learned how to add the RDS role, configure the RD Session Host server, and add RemoteApp applications. You've also learned how to deploy RemoteApp programs using Web Access, `.rdp` files, and Windows Installer files.

Although you've learned a lot so far, there's more. You'll also want to know how to manage these services after they've been installed.

Monitoring Remote Desktop Services

Once Remote Desktop Services is up and running, you'll need to monitor and manage it. Several RDS tools are available from the Start menu by selecting Start ➢ Administrative Tools ➢ Remote Desktop Services. The tools are as follows:

- Remote Desktop Services Manager
- Remote Desktop Session Host Configuration
- RemoteApp Manager
- Remote Desktop Web Access Configuration
- Remote Desktop Licensing Manager
- Remote Desktop Connection Manager
- Remote Desktops

Three menu items are available here without the Remote Desktop Services role installed. They are used to manage remote connections for administration and RDS. For example, Remote Desktops (covered in Chapter 14) is used to remotely administer clients and is included in a default installation. The other two items are Remote Desktop Services Manager and Remote Desktop Session Host Configuration.

The Remote Desktop Connection Manager, RemoteApp Manager, and Remote Desktop Web Access Configuration tools were covered earlier in this chapter.

Remote Desktop Services Manager

The Remote Desktop Services Manager is used to view information about users, sessions, and processes on a Remote Desktop Session Host server. You can also interact with sessions from this tool using Remote Control.

When you launch the Remote Desktop Services Manager from the computer hosting the RD Session Host server, the local server will automatically be added. However, if you manage more than one RD Session Host server, you can add all the servers to a single console. For large

environments, you can even group the RD Session Host servers using the My Group node in the console.

Users, Sessions, and Processes

When you're connected to an RD Session Host server, you'll have three tabs available. You can use these tabs to monitor and interact with activity on the server. Figure 25.21 shows the Remote Desktop Services Manager with the Processes tab selected.

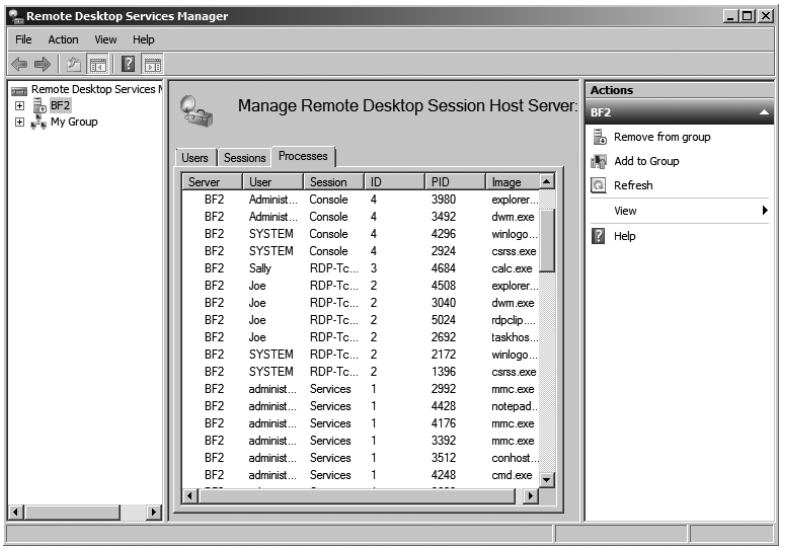


FIGURE 25.21 Remote Desktop Services Processes tab

In the figure, we have clicked the User header to order the list based on the user spelling. Sally is using a RemoteApp application (`calc.exe`), and Joe has a full desktop running. Notice that only one process is running for Sally, while Joe's session requires several supporting processes.

The three tabs are as follows:

Users This tab lists all the users who have sessions running on the server. It includes sessions that are active and disconnected.

Sessions The Sessions tab shows all the sessions for the server. It includes the RDS supporting sessions: Console, Services, and Listener. If any users connect, it will show their sessions as *RDP-TCP#x* (where *x* is the number assigned to the session).

Processes The Processes tab shows all the processes running on the server. You can right-click any process listed here and select End Process to kill it.

The Users and Sessions tabs give you many additional options to interact with sessions. If you right-click any of the sessions, you'll have the following choices:

Connect Allows you to connect to a user's session. When you connect to this session, the user will be disconnected.

This feature will work only when you access it from a Remote Desktop Services client session. It is disabled if you try to access it from the console session.

Disconnect Disconnects a user from an active session. Be nice, though. Send the user a message, and give them some time to clean up and log off before simply disconnecting them.

Send Message Sends a message to a session. The message will appear as a dialog box. The title will include who sent the message and when it was sent.

Remote Control Allows you to connect and interact with a remote session. This can be used to provide assistance to a user by either showing the user how to perform an action or watching and talking them through it. It is very similar to Remote Assistance, covered in Chapter 14, except that you have a lot more control with Remote Control than you'd have with Remote Assistance.

This feature will work only when you access it from a Remote Desktop Services client session. It is disabled if you try to access it from the console session.

Reset Deletes a session. Disconnected sessions still consume resources, so you can use this to delete a disconnected session and free up the server's resources.

Status Displays a status dialog box, as shown in Figure 25.22.

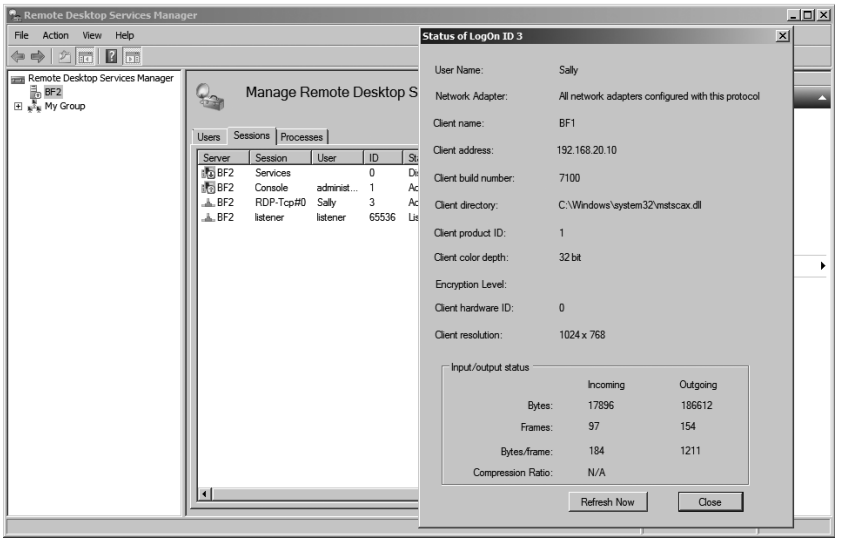


FIGURE 25.22 Session status from Remote Desktop Services Manager console

Figure 25.22 was launched by right-clicking the RDP-Tcp#0 session and selecting Status.

COMMAND-LINE TOOLS

In addition to the Remote Desktop Services Manager GUI, you can use several command-line tools to manage users, sessions, and processes in place of the Remote Desktop Services Manager, as shown in Table 25.2.

For more information about any of these tools, enter them from the command line with /? for help. Examples are given for each these with the assumption that a user with a username of Sally has an active session with a session ID of 1.

TABLE 25.2: Remote Desktop Services Manager Command-Line Tools

COMMAND	ACTION
`logoff`	Logs a user off from a session and deletes the session on the RD Session Host server. The number would the session ID number and can be obtained with `query session`. `logoff 1`

`msg`	Sends a message to a user on an RD Session Host server. The message will appear as a pop-up.
	`Msg Sally`
	`Message`
	`CTRL + Z Enter`

`query process, qprocess`	Displays information about processes running on an RD Session Host server. No arguments are needed. `query process`
`query session, qwinsta`	Displays information about sessions running on an RD Session Host server. No arguments are needed. This can be used to identify the session ID, the username, and the session name of all sessions. `query session`

`query users, quser`	Displays information about user's sessions running on an RD Session Host server. This can be used to determine whether the session is active, how long it's been idle, and when the user logged on. If executed without arguments, it shows all information on all users. If executed with the name of an active user, it shows only that user's information.
	`query user`
	`query user Sally`

`Tsdiscon`	Disconnects an active session on an RD Session Host server. `Tsdiscon 1`
`Tscon`	Connects to a disconnected session on an RD Session Host server. `Tscon 1`
`Tskill`	Ends a process running in a session on an RD Session Host server. Processes can be identified with the query process command. `Tskill mstsc`

Remote Desktop Session Host Configuration

You can use the Remote Desktop Session Host Configuration console to configure many of the settings for your RD Session Host server. Settings in this console will affect all the users who connect to the server. Figure 25.23 shows the configuration console.

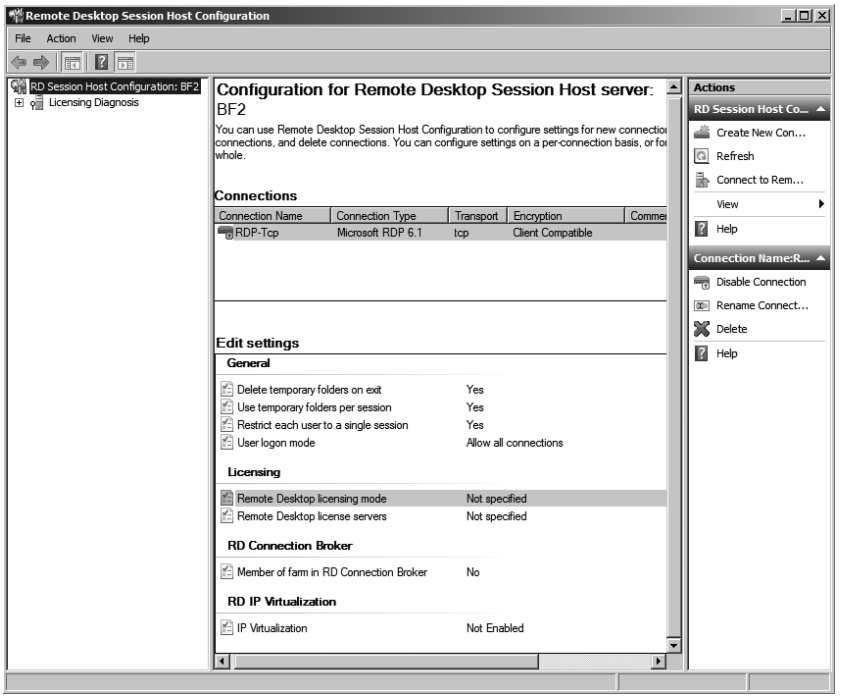


FIGURE 25.23 Remote Desktop Session Host Configuration console

There are three major types of settings you can configure with the majority of the server configuration done through the RDP-Tcp Connection property page.

RDP-Tcp Connection settings You can use the RDP-Tcp Connection properties to configure all the connections to the RD Session Host server. This includes security settings, session settings, remote control settings, and more. The majority of the configuration for the RD Session Host server is done through these properties.

Edit Settings The Edit Settings section shows the current settings for four additional areas. If you double-click any of the areas, you can see the properties sheet with the four tabs that can be used to supplement the RDP-Tcp Connection settings.

Licensing Diagnoses If you are receiving errors related to RDS licensing, you can use the Licensing Diagnoses tool to help you identify the problem. Select this in the tree pane on the left.

RDP-Tcp Connection

You can view and modify the properties of RDP-Tcp Connection by either double-clicking it or right-clicking it and selecting Properties. The properties sheet includes eight tabs.

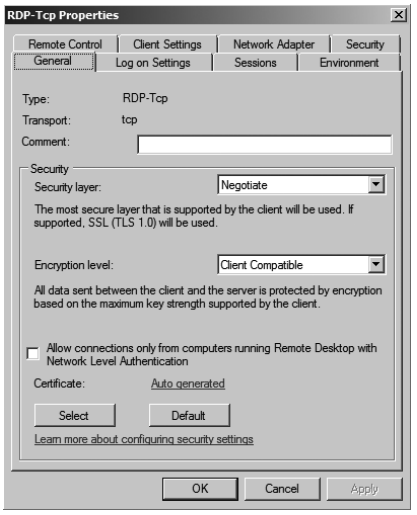
This connection is available even if the Remote Desktop Services Session Host role has not been installed. Before the role is added, this connection will allow two connections for administrator purposes. When the role is added, it is changed to allow unlimited connections.

You can add connections if your server includes multiple network adapters.

RDP-Tcp Properties General Tab

Figure 25.24 shows the General tab. You can add a comment here that may be useful if you have multiple NICs and multiple connections you're using on your RD Session Host server. However, the primary use of this page is to configure security.

FIGURE 25.24
RDP-Tcp Properties General tab



RDS supports both the RDP Security Layer and SSL (TLS 1.0). SSL (TLS 1.0) is more secure than RDP Security Layer. If the Security Layer is set to Negotiate (as shown in the figure), the RDS server will attempt to use SSL (TLS 1.0) first. If the client doesn't support it, it will use RDP Security Layer instead, which provides weaker security.

Earlier, single sign-on was mentioned, and this is one of the settings you need to verify to support single sign-on. It must be set to Negotiate or SSL (TLS 1.0). You'll also need to verify the "Always prompt for password" option is not selected on the Log On Settings tab.

Additionally, you'll need to use a certificate to use SSL (TLS 1.0). If you installed RDS using the exercises in this chapter, an autogenerated (self-signed) certificate was created and added.

SELF-SIGNED OR TRUSTED CERTIFICATE

Although you can create a self-signed certificate, Microsoft recommends you obtain a certificate from a trusted certificate authority (CA) for better security. This trusted CA can be a public one such as VeriSign or Thawte or an Active Directory Certificate Services server built internally. However, for small organizations where the server is used internally only, you can use a self-signed certificate without any problems.

You can select from one of four encryption levels. This can encrypt the data sent to and from the server to prevent sniffing attacks. The choices are as follows:

Low Data sent from the server to the client is not encrypted. Data sent to the server from the client is encrypted using 56-bit encryption.

Client Compatible Data is encrypted to and from the server using the maximum key strength supported by the client. This is the default setting.

High Data is encrypted to and from the server using 128-bit encryption. Clients that don't support 128-bit encryption can't connect.

FIPS Compliant Data is encrypted to and from the server using Federal Information Process Standard (FIPS) 140-1 validated encryption methods. FIPS is a series of documents published by the National Institute of Standards and Technology (NIST). When this is selected, clients that don't support FIPS 140-1 encryption can't connect.

RDP-Tcp Properties Log On Settings Tab

You can configure what credentials are used for sessions through the Log On Settings tab of the RDP-Tcp properties sheet. A user always has to provide their own credentials to determine whether they should be able to access the server, but you can use this page to alter the credentials used for the session.

Figure 25.25 shows the Log On Settings tab.

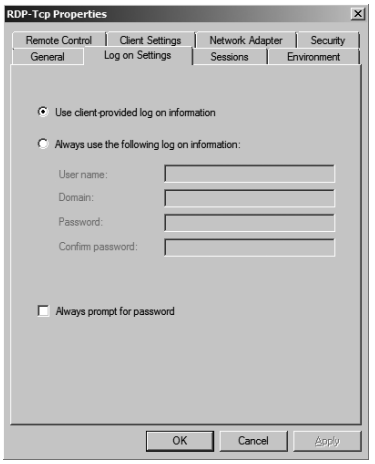


FIGURE 25.25
RDP-Tcp Properties dialog box's Log On Settings tab

The default is to use the client-provided logon information. However, you could also create an account with specific permissions and privileges on the RD Session Host server. Then, when users connect and authenticate, the session will start with the credentials you provided. This can be useful if you're hosting an application with special rights and permissions.

The "Always prompt for password" setting has two possible uses. First, if you want to configure single-sign-on as discussed earlier, you would ensure that this box is deselected and the security layer (on the General tab) is set to either Negotiate or SSL (TLS 1.0). However, if your clients frequently access the RDS server from public places and you want to add another layer of security, you can select this box. It will force users to always provide a password even if they've configured their password to be saved. This prevents an attacker from launching an RDS session if a valid user leaves their system unlocked. The attacker will be prompted for a password. As long as the user didn't write down their password on a little yellow sticky attached to the monitor, the attack is thwarted.

RDP-Tcp Properties Sessions Tab

The Sessions tab can be used to override user settings for how to handle disconnected sessions, active session limits, and idle session limits. By default, these settings are configured on a per-user basis using Active Directory Users and Computers.

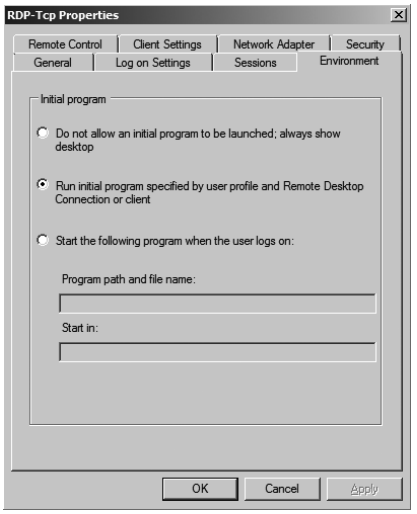
However, if you want all users who connect to the server to have the same settings, you can use this page to override the individual settings. This tab was covered in more depth in Chapter 14.

RDP-Tcp Properties Environment Tab

The Environment tab can be used to launch a specific application when a user connects. It's very common to use an RD Session Host server to host a line-of-business application. If you're specifically using RDS to host an in-house application, it makes a lot of sense to launch the app as soon as the user connects.

Figure 25.26 shows the Environment tab. The default setting is shown. You can override this for every user by either specifying that applications should not be launched or identifying a specific application to run when the user logs on.

FIGURE 25.26
RDP-Tcp Properties Environment tab



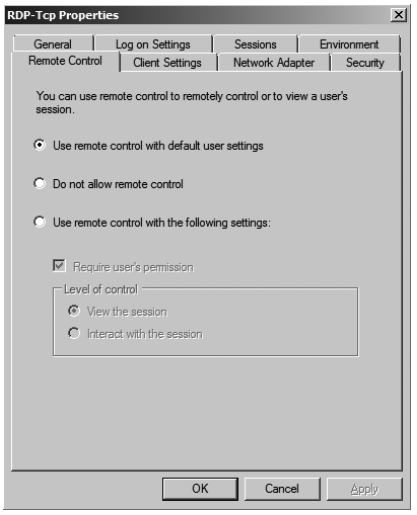
To specify a starting application, you simply provide the program path and filename of the application. Some applications require the starting path to be specified so that the application can access specific application data. If necessary, you can specify the path in the Start In text box.

RDP-Tcp Properties Remote Control Tab

Remote Control is a neat feature available with an RD Session Host server. As mentioned earlier, an administrator can use it to interact with a user's session to either show a user how to accomplish a task or talk a user through the task while observing the actions on the screen.

Figure 25.27 shows the Remote Control tab. The default setting is shown using the default user settings. You can also completely disable remote control or configure remote control with special settings that apply to all users connecting to the server.

FIGURE 25.27
RDP-Tcp Properties Remote Control tab



When configuring server settings for remote control, you can set it to require the user's permission or not. Additionally, you can configure the level of control to either view the session or interact with the session.

If your company is managing an RD Session Host server, there's nothing wrong with setting it to not require the user's permission in many instances. Although it makes sense to require the user's permission in a peer-to-peer Remote Assistance scenario, it's different when users are connecting to a corporate RDS server.

The user (an employee within the company) is asking for help, and the help-desk professional (another employee within the company) is there to provide assistance. Requiring the help-desk professional to request permission from the employee to connect is often just an extra step that isn't required. Of course, if employees may be accessing sensitive data that the help-desk professional shouldn't see, then requiring the user's permission to connect is appropriate.

If you do set it so that the user's permission is not required, you may want to provide some type of notification to the user that their sessions may be monitored. Many companies provide this notification in an acceptable use policy.

RDP-Tcp Properties Client Settings Tab

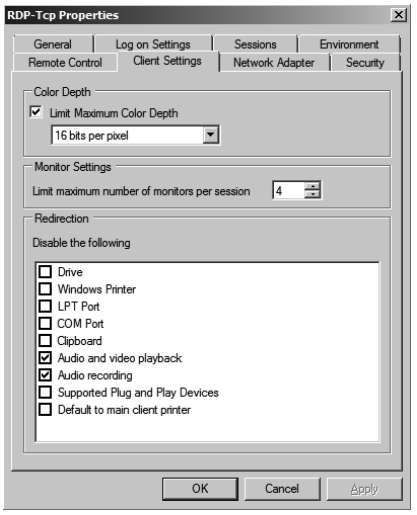
The Client Settings tab is useful if your users are experiencing performance issues. You can reduce some of the capabilities to provide better performance.

For example, you can reduce the color depth if users are connecting over a slow connection. The different settings are 15 bits, 16 bits, 24 bits, or 32 bits per pixel. For most users and most applications, the reduced color depth may not be noticeable, while the increased speed will be greatly appreciated.

Figure 25.28 shows the Client Settings tab. Notice that you can also disable redirection for several devices from this page.

Redirection allows users to access local resources in the remote session. For example, a user may want to be able to access files on their local C drive on their system. With the check box deselected (not selected to *disable* redirection), they can configure redirection.

FIGURE 25.28
RDP-Tcp Properties Client Settings tab



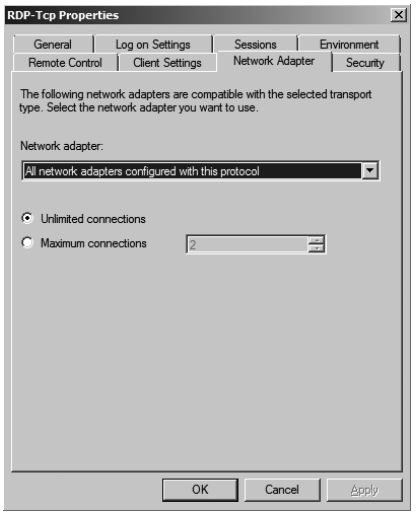
A key point is that this page is used to disable redirection on a global scale. If redirection is not disabled, users have the ability to select or deselect redirection for individual items on a per-connection basis. If you refer to Figure 25.18 earlier in this chapter, it shows that the user has several choices for redirection. Users have similar choices if they connect with Remote Desktop Connection.

RDP-Tcp Properties Network Adapter Tab

If your RD Session Host server is multihomed, you can configure which network adapters will be used for the RDP-Tcp connections.

Figure 25.29 shows the Network Adapter tab. In the figure, it's set to use all network adapters, but if you select the drop-down box, you'll see that you can select individual NICs.

FIGURE 25.29
RDP-Tcp Properties Network Adapter tab



When the server is configured as an RD Session Host server, it is set to "Unlimited connections." You can also use the "Maximum connections" setting to limit the number of connections the server will accept. If you find that an RDS server functions best below a certain number of connections, you could configure the maximum connections to this threshold.

You are still legally limited to the number of licenses you've purchased for the server. If you're using per-user CALs, the license server doesn't track the CALs, but you can configure the maximum connections on this page to coincide with the number of licenses you've purchased.

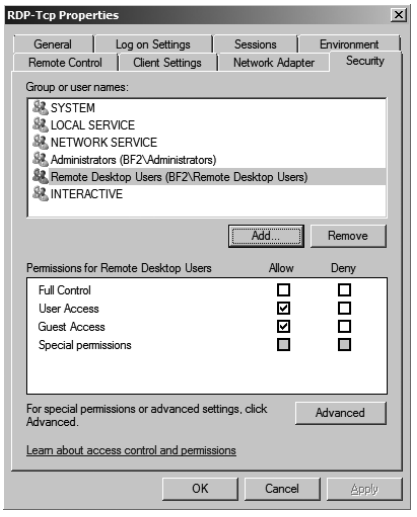
Before you configure a server as an RD Session Host server, the "Maximum connections" setting is set to 2. If Remote Desktop for administration is enabled, the server will support a maximum of two connections.

RDP-Tcp Properties Security Tab

The Security tab allows you to modify permissions granted to users (see Figure 25.30). As soon as you select this tab, a dialog box appears reminding you to use the local Remote Desktop Users group to control who can log onto the RD Session Host server.

In other words, you only need to use this tab to modify advanced permissions for a special group. For example, you may have a group of RD administrators that need to be able to do anything on your RD Session Host server. You could use a Windows Global group to organize the users, add them to the Security page, and allow Full Control permissions.

FIGURE 25.30
RDP-Tcp Properties Security tab



The Security tab includes four permissions:

Full Control Full Control includes the following permissions: query information, set information, remote control, logon, logoff, message, connect, disconnect, and virtual channels.

User Access User Access includes the following permissions: query information, logon, and connect.

Guest Access Guest Access includes only the Logon permission.

Special permissions Any of the following special permissions can be individually allowed or denied: query information, set information, remote control, logon, logoff, message, connect, disconnect, and virtual channels.

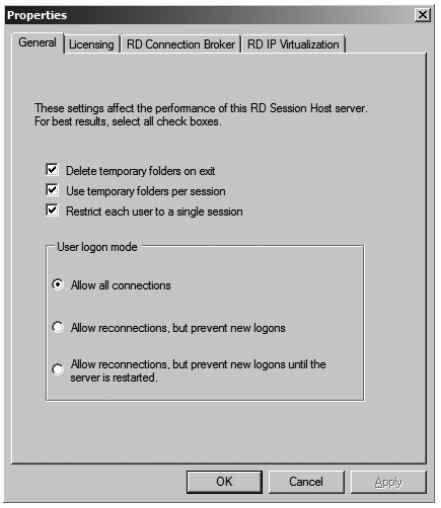
Edit Settings

The Edit Settings property page includes four tabs. You can access any of these settings by double-clicking any of the settings in the General, Licensing, RD Connection Broker, or RD IP Virtualization sections.

General Tab

Figure 25.31 shows the General tab. It's recommended to keep all the check boxes selected for the best performance of the server. Notice the last check box prevents users from opening more than one session—this refers to full desktop sessions, not RemoteApp applications. Users will be able to launch multiple RemoteApp applications with this selected.

Figure 25.31
Edit Settings
General tab

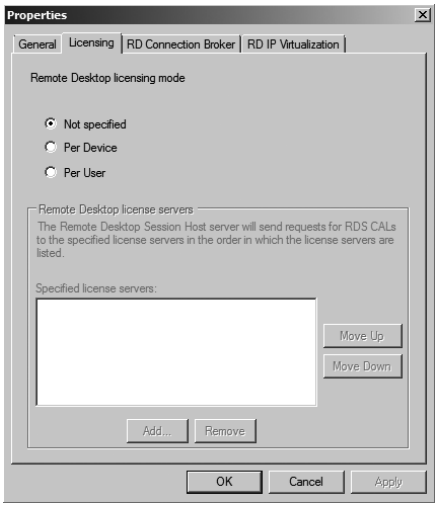


Licensing Tab

The Licensing tab allows you to choose between Per Device or Per User. As a reminder, it's recommended to postpone configuring a licensing server until your RD Session Host servers are up and running. Figure 25.32 shows this tab.

Before the 120-day grace period, you'll need to revisit this page and set the licensing mode. When you select either Per Device or Per User, you'll also need to specify the license server. In very large organizations, you can use multiple licensing servers. A single licensing server can manage licenses for multiple RDS Session Host servers.

FIGURE 25.32
Edit Settings
Licensing tab



RD Connection Broker Tab

RD Connection Broker is needed only if you have more than one RD Session Host server. The RD Connection Broker provides two important features:

Load balancing If you have multiple RD Session Host servers, you can add the servers to a Connection Broker farm. When a user connects, the RD Connection Broker will determine which server has the lightest load and will redirect the connection to that server.

Reconnects users to the correct session If a user becomes disconnected from a session, the RD Connection Broker will ensure they are connected back to the same session on the original server. For example, say that Sally is connected to BF2 but the network has a problem and disconnects her. When she reconnects, the Connection Broker recognizes she has an active session on BF2 and will redirect her connection to that server.

RD IP Virtualization Tab

If an application requires each connection to have a separate IP address, you can use RD IP Virtualization. Normally, every session will have a single IP address. Although this works for the majority of applications, there are a few instances where separate IP addresses are required.

RD IP Virtualization also requires a DHCP server to be configured to provide virtual IP addresses.

LICENSING DIAGNOSIS

The last tool you have available in the Remote Desktop Session Host Configuration console is Licensing Diagnosis. When licensing issues crop up, they've been challenging to resolve in past versions of Windows and Terminal Services. This tool is a welcome addition.

Figure 25.33 shows some of the information provided from the Licensing Diagnosis console.

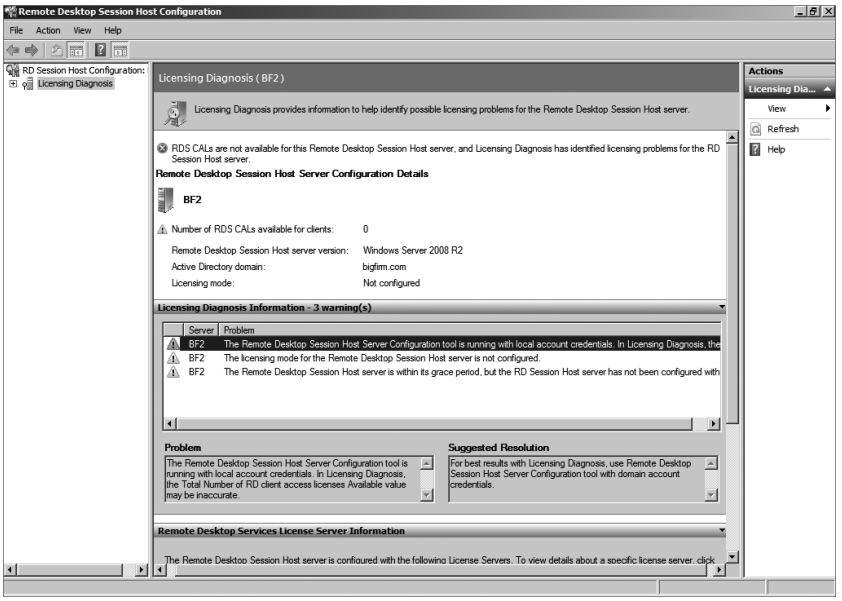


FIGURE 25.33 Licensing Diagnosis tab

In the figure, licensing hasn't been configured yet, and RDS CALS have not been added. However, by reviewing the entries in the center panes, the issue is easy to identify. This tool becomes an easy reference to identify any licensing issues.

Remote Desktop Licensing Manager

Although you have a grace period when RDS will function normally, after the grace period ends, RDS will no longer accept connections if licensing is not configured. The grace period lasts for 120 days or until the first permanent RDS CAL is issued by a license server, whichever occurs first.

As mentioned previously, you can choose between per-user or per-device Remote Desktop Services Client Access Licenses (RDS CALs). The licensing server must first be activated before you can install the licenses.

After you've configured your RDS environment, you'll want to configure the license server. The RD Licensing Manager is used to install, issue, and track the availability of RDS CALs on a Remote Desktop license server. Licenses are purchased through a variety of different methods, depending on your company's relationship with Microsoft, such as the following:

- Enterprise Agreement
- Campus Agreement
- School Agreement
- Services Provider License Agreement
- Other Agreement

If you have one of these agreements with Microsoft, the best way to obtain licenses is through this agreement. It's also possible to purchase licenses through retail channels by

purchasing a license pack. For detailed information on how to purchase licenses, check out this page: `http://technet.microsoft.com/library/cc786167.aspx`.

The license server can be on the same server as the RD Session Host server, or for larger implementations of Remote Desktop Services with multiple servers, a single license server will manage licenses for multiple RDS servers.

Older Terminal Services license servers used a discovery scope to allow TS servers to locate the license server. If you're installing the license server on Windows Server 2008 R2, this is not needed. Instead, you should use the Remote Desktop Session Host Configuration tool to specify a license server for the RD Session Host server to use. This is done on the Licensing tab of the RDP-Tcp Connections Properties dialog box where you identify the type of RDS CALs used for the server (per user or per device).

If you've performed the steps in this chapter to install and configure an RD Session Host server, you can configure the RD Licensing Manager by following these steps:

1. Launch the RD Licensing Manager by selecting Start ➢ Administrative Tools ➢ Remote Desktop Services ➢ Remote Desktop Licensing Manager.
2. Click the plus (+) to expand All Servers, and you'll see your server marked with a white *X* in a red circle.
3. Select your server. Right-click your server, and select Activate Server.
4. Review the information on the wizard's Welcome page, and click Next.
5. On the Connection Method page, accept the default of Automatic Connection (Recommended). Use this method if the RDS server has access to the Internet. If the server doesn't have access to the Internet, you can connect with another computer over the Internet or via a telephone. Click Next.
6. The Company Information page will appear. Enter your first name, last name, company, and country. This information is used if you need help from Microsoft. Click Next.
7. Enter the additional information requested on the Optional Company Information page. Click Next.
8. A dialog box will appear with a progress bar. The server is connecting to the Microsoft Clearinghouse and is being activated. When it completes, the completion page will appear.
9. Deselect the Start Install Licenses Wizard Now check box, and click Finish. At this point, the licensing server is activated, but there aren't any RDS CALs installed.

Set Per User or Per Device

It may be necessary to return to the Remote Desktop Session Host Configuration console and set the Remote Desktop licensing mode. After launching the console, double-click the Remote Desktop licensing mode to access the property page. Select Per Device or Per User depending on what type of licenses you have purchased, and enter the name of the license server.

10. Right-click your server, and select Install Licenses. This will launch the wizard to install your licenses. There are multiple paths this can take, depending on what type of licenses you've purchased and where you've purchased them from.

The Bottom Line

Limit the maximum number of connections You can limit the maximum number of connections for the server for performance reasons or to help ensure you remain compliant with the licensing agreement.

Master It You want to limit the maximum number of connections to 100. How can you do this?

Add an application to an RD Session Host server Once the RDS role is added and the RD Session Host server is configured, you can add applications to make them available to the server.

Master It Your company has purchased an application that supports multiuser access. You want to install it on the RD Session Host server. What should you do?

Add a RemoteApp for Web Access RemoteApp applications can be configured so that they are accessible to users via a web browser. Users simply need to access the correct page and select the application to launch it.

Master It Assume you have already configured your environment to support RemoteApp applications. You now want to add a RemoteApp application. What should you do?

Add a RemoteApp to the Start menu RemoteApp applications can be configured so that they are accessible to users from the Start menu of their system. Once configured, users simply select the item from their Start menu to launch it.

Master It Assume you have already configured your environment to support RemoteApp applications. You now want to add a RemoteApp application so that it is accessible to users via the Start menu. What should you do?