

# Remote desktop



**ASSIGN  
BUSTER**

Ever wished you could access your PC from the road? With Remote Desktop in Windows 7, you can. Remote Desktop connects two computers over a network or the Internet. Once connected, you'll see the remote computer's desktop as if you were sitting right in front of it, and have access to all its programs and files. This feature is included with all editions of Windows 7, but you can only connect to computers running the Professional, Ultimate, or Enterprise editions. Use Remote Desktop to access one computer from another remotely.

For example, you can use Remote Desktop to connect to your work computer from home. You will have access to all of your programs, files, and network resources, as if you were sitting in front of your computer at work. While you are connected, the remote computer screen will appear blank to anyone at the remote location who sees it.

#### 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](http://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 <https://assignbuster.com/remote-desktop/>

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 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 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](http://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.sp](http://www.hobsoft.com/products/connect/jwt.sp), and there is a free Linux RDP client available at [www.rdesktop.org](http://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

<https://assignbuster.com/remote-desktop/>



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