



realtimepublishers.com[™]

The Definitive Guide[™] To

Windows Server 2003 Terminal Services

Updated Edition



Greyson Mitchem

Chapter 8: Managing and Optimizing Terminal Services Performance and Capacity	179
Common Bottlenecks.....	179
Windows System Resource Manager	179
Installing WSRM	179
Configuring WSRM.....	180
Controlling Leaky Applications with WSRM	182
Third-Party Products for Server Resource Management	183
triCerat Simplify Resources.....	183
Citrix Presentation Server	184
Virtualization and Terminal Services	185
Creating a Virtual Hosting Environment	185
Managing Virtual Server Resources	187
Implementing Terminal Services in a Virtual Environment.....	188
Why Virtualize?	188
Application Isolation.....	189
Summary	190
Appendix A: Terminal Services Clients	191
Appendix B: Important URLs.....	192
Appendix C: Registry Changes.....	193
Appendix D: Script Reference	194
USRLOGON.CMD.....	194
TSSHUTDOWN Wrapper.....	196
Maintenance Reboot Script.....	198
Remote Desktop Web Connection Experience Dropdown.....	199
Appendix E: Terminal Services Command Line Reference.....	201

Copyright Statement

© 2006 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

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

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

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

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

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

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 8: Managing and Optimizing Terminal Services Performance and Capacity

You now have the tools you need to design and build a terminal server infrastructure, install applications, secure the OS to protect from worms, viruses, and even the users themselves, and make the desktops and applications available to designated users. This chapter will cover methods used to optimize your terminal servers so that you can support the largest possible number of users on the least possible number of servers—doing more with less. The chapter will start by looking at optimization and management options for each server as an individual unit, then discuss the latest trend in the server world—virtualization—and how it fits into a Terminal Services environment.

Common Bottlenecks

When designing any computing environment, your goal should be to maximize server utilization while minimizing the impact on user experience when resources become constrained. To this end, you need to be concerned with eliminating bottlenecks—server resources that become constrained before all others—and preventing processes and applications from taking more resources than they legitimately need.

The most common bottlenecks in a terminal server environment are memory, CPU utilization, and disk I/O. Chapter 3 covered the various RAID options available to optimize your hard disk performance on an I/O-intensive terminal server. This section will focus on processor and memory management.

Windows System Resource Manager

Windows System Resource Manager (WSRM) is a free add-on to WS2K3 Enterprise and Datacenter editions. WSRM manages the allocation of processor and memory resources based on a rule set that you can configure. You can define resource priorities based on process name or user/group name, thus ensuring that no single process or user can consume all available resources on your terminal server, impacting other users.

Installing WSRM

WSRM comes on the CD-ROM included with WS2K3 Enterprise and Datacenter editions. If you do not have the CD-ROM, it can be downloaded in ISO format from Microsoft at <http://www.microsoft.com/technet/downloads/winsrvr/wsrp.msp>. Once you have the CD-ROM, run the setup program for your server architecture (x86, AMD64 or IA64) to install WSRM. No reboot is required.

Configuring WSRM

WSRM has both a GUI and command-line interface. Figure 8.1 shows the WSRM administrative tool.

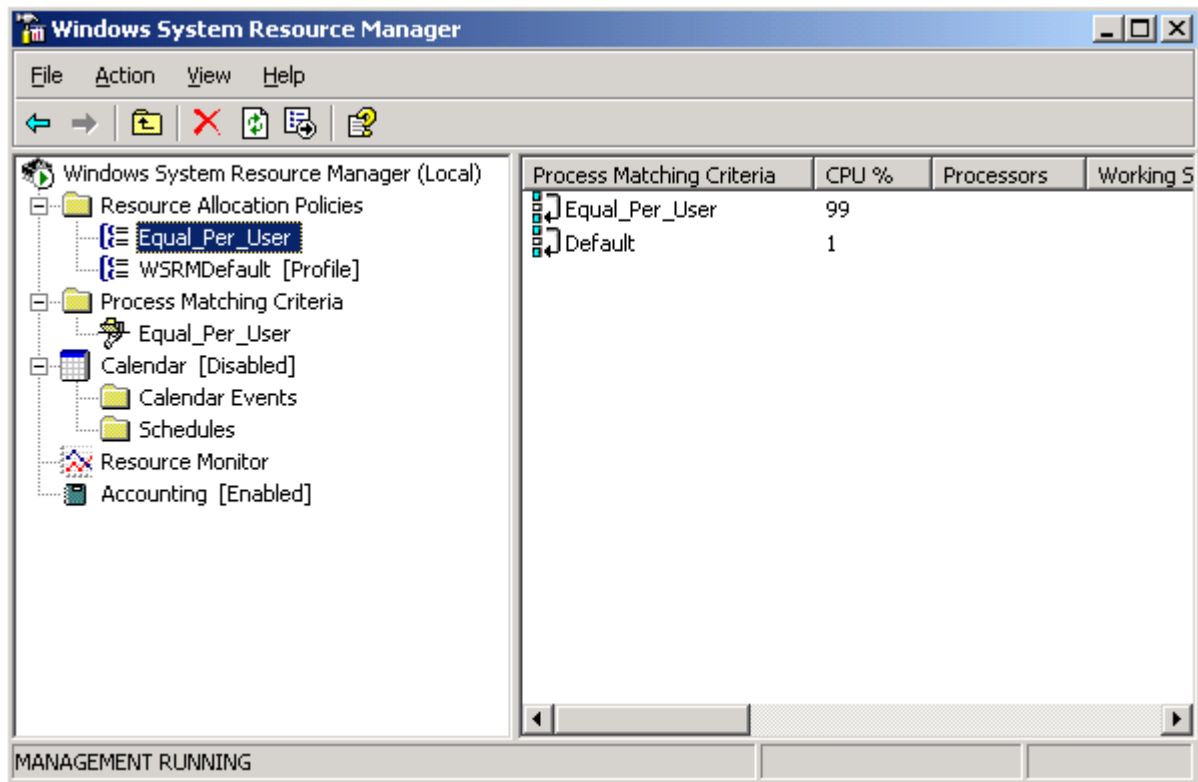


Figure 8.1: The WSRM administrative tool.

WSRM allows you to create multiple Resource Allocation Policies. Each policy defines a set of rules used to allocate server resources. For example, you could specify that Outlook.EXE is entitled to 50percent of available resources, and Winmine.exe (the Mine Sweeper game) is only entitled to 1 percent. This way, one user's game cannot slow another user's Outlook session.

Only one Resource Allocation Policy can be in effect at a time. You can either manually toggle between policies via the admin tool, or you can use the built-in calendar function to change which Resource Allocation Policy is in effect at different times of day. This way, you can provide maintenance or batch processes more resources at night and user applications more resources during the day. You can also set up rules that will terminate applications if they try to consume more than their allocated share of resources—great for leaky applications on a terminal server.

☛ Unfortunately, the process termination rules can only be based on memory utilization, not CPU utilization, so although you can suppress a process's allocation of CPU; you cannot actively terminate a runaway process.

The WSRM ships with two built-in rules—WSRMDefault and Equal_Per_User. If your terminal server users all run the same applications and you do not need to favor one user over another, the Equal_Per_User rule may be all that you need to implement. This rule shapes CPU and memory utilization such that each user on the terminal server receives an equal share of available resources.

If, however, you want to allocate more resources to specific applications or specific users, you first must create additional Process Matching Criteria. To do so, right-click the Process Matching Criteria node, and select New. In the dialog box that Figure 8.2 shows, enter either application names, paths with wild cards, or user and groups running processes.

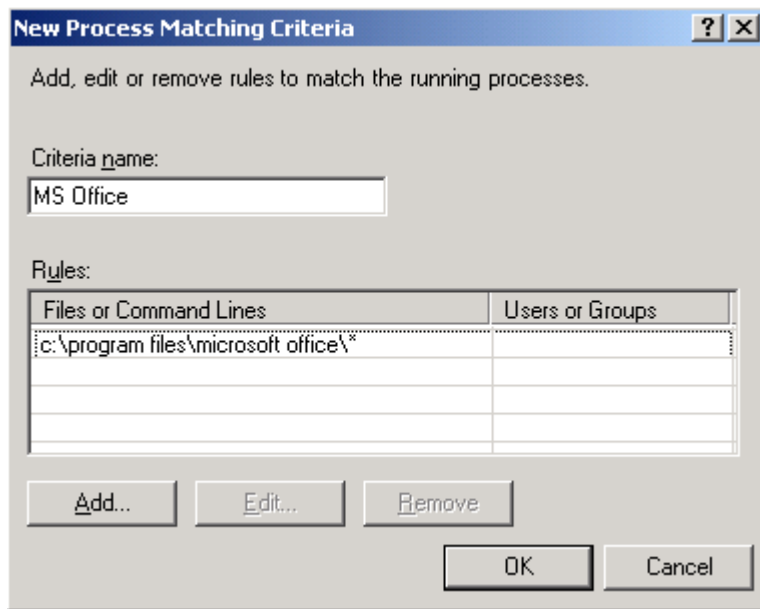


Figure 8.2: Creating a Process Matching Criteria in WSRM.

The rule that Figure 8.2 shows would match all EXEs in the Microsoft Office directory. Once you create the criteria sets you want to filter on, you then create a new Resource Allocation Policy by right-clicking that node, and selecting New. When creating a new Resource Allocation Policy, you grant percentages of available resources to various Process Matching Criteria sets. Figure 8.3 shows a policy called Favor_Office, which grants Office applications as much as 70 percent of available resources and restricts Internet Explorer (IE) to only 20 percent.

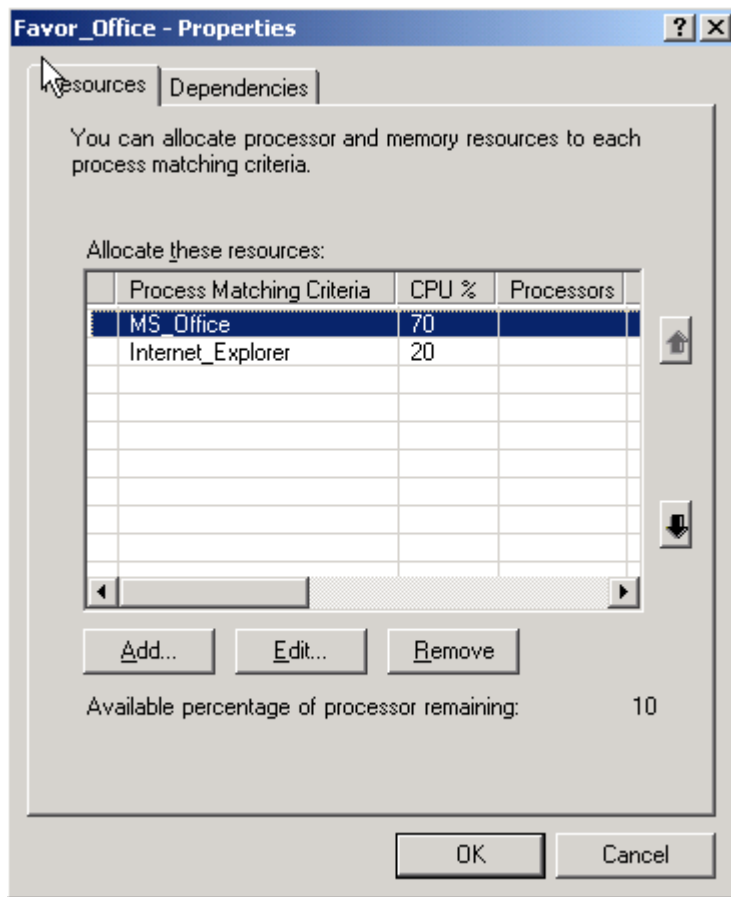



Figure 8.3: Creating a new Resource Allocation Policy in WSRM.

 Be sure to always leave a percentage unallocated for any processes that may be run that do not match any of your criteria.

Keep in mind that WSRM uses “soft limits”. In the earlier example, if no one is running any Office applications on the terminal server, IE will be allowed to consume as much processor as it wants. However, once any user launches an Office application, IE will be throttled back if the processor becomes constrained.

Controlling Leaky Applications with WSRM

When adding a Process Matching Criteria to a Resource Allocation Policy, you also have the option to limit the amount of committed memory that any process matching the criteria can consume. When limiting committed memory, you have the option to either terminate the process when the limit is crossed or simply log an event log message. Figure 8.4 shows how to limit a Process Matching Criteria to only 30MB of memory and terminate the process if it exceeds that limit.

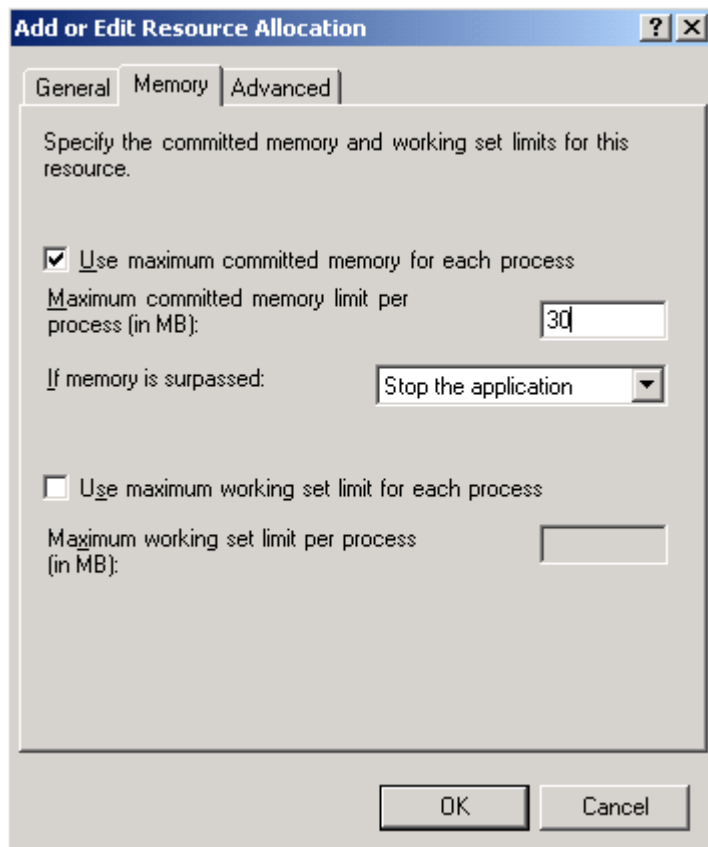


Figure 8.4: Limiting the maximum committed memory for a Process Matching Criteria.

Third-Party Products for Server Resource Management

In addition to WSRM, there are third-party products available to help you manage resources on your terminal servers. These tools are often more tuned to the specific needs and configurations of a terminal server environment than WSRM, which is designed to fit a broad range of server types.

triCerat Simplify Resources

Simplify Resources is part of triCerat's Simplify Suite, which is considered the standard for managing user profiles, printing, user environment, and resource management in a Terminal Services environment. Simplify Resources overcomes application performance problems by taking control of the way Windows allocates and manages CPU and memory resources. It dramatically improves the user experience and increases the number of sessions that can be effectively supported by your servers. Simplify Resources also provides full DLL rebasing technology and real-time system monitoring and reporting capabilities, tracking performance across an entire farm or per server, session, and application. Figure 8.5 shows the Simplify Resources admin console.

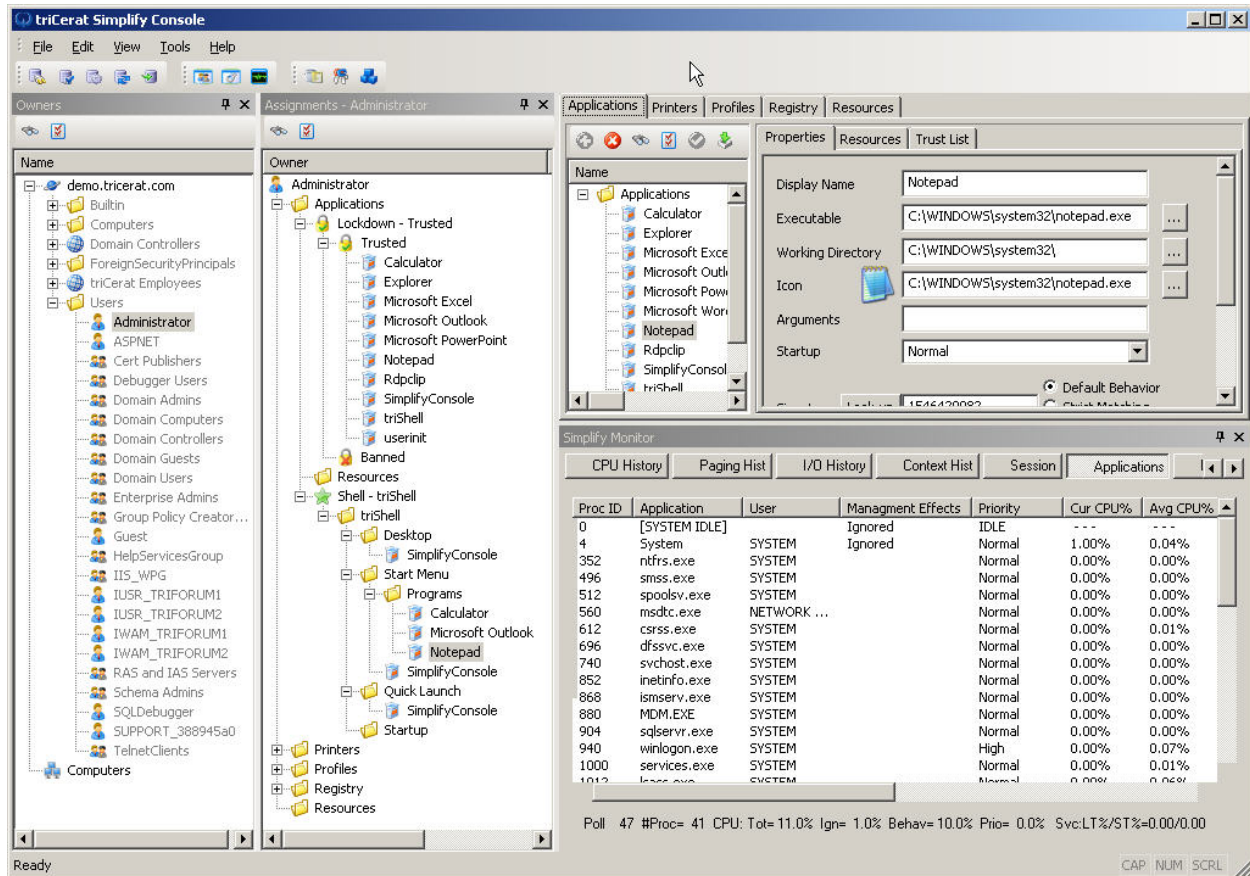


Figure 8.5: triCerat's Simplify Resources admin console.

Citrix Presentation Server

Citrix Presentation Server is far more than just a resource management tool, as it also replaces Session Directory features and enhances Terminal Services with application publishing abilities. From a resource management perspective, Presentation Server 4.0 includes a number of features to keep your terminal servers running at their maximum capacity. Presentation Server includes both a processor management subsystem, similar to WSRM's Equal_Per_User policy, but it is not configurable beyond that. It also includes a memory optimization subsystem that detects DLLs that are not properly configured to be available to all user sessions and adjusts them accordingly.


Virtualization and Terminal Services

Virtualization allows you to run multiple, independent OSs within a single physical server. The CPU, disk, and memory of the host system are shared among all virtual machines running on it. How does virtualization fit into a Terminal Services environment? In a number of different ways:

- Virtualization can isolate incompatible applications without wasting the cost and resources of an entire physical server.
- Virtualization enables you to better utilize the available resources of a large physical server by breaking it into smaller OS chunks.
- Virtualization can be used to serve independent desktop environments to users via Terminal Services.

Creating a Virtual Hosting Environment

There are two major virtual machine platforms available today—Microsoft’s Virtual Server and EMC’s VMWare ESX and GSX. Each of these products has its own feature set and limitations, so the one you select will depend on your unique requirements. VMWare ESX is unique among these products as it is an OS unto itself—a small LINUX microkernel. This allows you to minimize the resources taken up by the host OS on the server. Both Virtual Server and VMWare GSX run within WS2K3, so the host OS is a full install of Windows with its own CPU and memory requirements.

 Both EMC and Microsoft also offer a “workstation”-class version of their virtual machine software—Microsoft Virtual PC and VMWare Workstation. Other than support for enterprise-class hardware and recovery/migration tools, the main difference between these products and their server equivalents is that they run as applications, not services, so the hosting environment must be run under the context of a user session.

Keeping with the Microsoft focus, this guide will use Virtual Server as an example hosting environment. You can download a 180-day evaluation of Microsoft’s Virtual Server 2005 R2 from

<http://www.microsoft.com/windowsserversystem/virtualserver/evaluation/trial/default.msp>

Before installing Virtual Server, you must first enable the Application Server role. Virtual Server’s administration tool is a Web interface and is served via IIS. If you have a large number of servers running Virtual Server, you can elect to install only the Web interface on a few of them, reducing the number of IIS installations you must perform.

Virtual Server installs as a Windows Service and does not require a reboot. Once the installation is complete, you will see a README Web page listing the administrative console URL as well as some information on creating your virtual machines.

Click the admin console hyperlink or access it via the start menu to create your first virtual machine. On the left pane of the Web console, click Create under Virtual Machines. Doing so will open the dialog box that Figure 8.6 shows.

Create Virtual Machine

Virtual machine name
Type the name for the virtual machine file to create a virtual machine in its own folder saved in the default configuration folder specified on the [Virtual Server Paths](#) page. To create a virtual machine in a different location, provide a fully qualified path.

Virtual machine name:

Memory
The amount of memory can be from 4 MB through 818 MB (736 MB maximum recommended).

Virtual machine memory (in MB):

Virtual hard disk
Before you can install an operating system on this virtual machine, you must attach a new or existing virtual hard disk to it. A virtual hard disk is a .vhd file that is stored on your physical hard disk and contains the guest operating system, applications and data files.

Create a new virtual hard disk
This option creates an unformatted dynamically expanding virtual hard disk in the same directory as the virtual machine configuration file. The maximum size allowed is 127 GB.

Size: Units:

Figure 8.6: Creating a new Virtual Machine in Virtual Server.

In this window, you can specify the name of the virtual machine, the amount of memory that the virtual machine should have, and the size of the virtual hard disk to create. The name is a label only and does not have to match the host name of the system once the OS is installed. You can also change the memory allocation at any point (with the virtual machine shut down).

When creating a virtual hard disk, there are a number of options available:

- **Dynamically Expanding**—The size of the virtual hard disk expands as data is written to it. The initial size is typically less than 100KB, but as data is added, the disk size will expand until it reaches the limit you specified when the disk was created. Virtual Server will monitor the available hard disk space on the physical computer and notify you if the dynamically expanding disk starts to approach the limits of available space on the physical computer.
- **Fixed-size**—The size of the virtual hard disk is fixed to the size you specify when you create the disk. The initial size and the limit are equal; when you create the disk, it immediately uses the amount of space you specify. For example, if you create a 10GB fixed-size virtual hard disk, it will immediately use 10GB of space.

- **Differencing**—A differencing virtual hard disk is a virtual hard disk associated with another virtual hard disk in a parent-child relationship. The differencing disk is the child and the associated virtual disk is the parent. The parent disk can be any type of virtual hard disk. The differencing disk (the child) stores a record of all changes made to the parent disk and provides a way to save changes without altering the parent disk. In other words, by using differencing disks, you ensure that changes are made by default to the differencing disks and not to the original virtual hard disk. You can, however, elect to merge changes from the differencing disk to the original virtual hard disk when it is appropriate to do so. (Note: In Virtual PC, these are called “Undo Disks”.)
- **Linked**—Linked disks are intended for converting a physical disk to a virtual hard disk. You cannot use a linked disk as a startup disk. Virtual machines will not be able to start with linked disks attached. If you turn on a virtual machine with a linked disk attached, the virtual machine will turn itself back off and write an event to the event log.

When creating a new virtual machine, the default is to use a Dynamically Expanding disk. If you want to use one of the other types, you must create the virtual disk first and then specify it in the New Virtual Machine dialog box. Differencing disks are useful for testing a QA work, as they enable you to “roll back” to the default state at any time. Some systems administrators have experimented with using Differencing disks for production terminal servers, and rolling back changes on a regular basis as a way of keeping the terminal servers in a “like new” state, but the disk and processor overhead that this creates for the host system can be cost prohibitive.

Managing Virtual Server Resources

In addition to specifying the amount of memory that you want to allocate to each virtual machine, you can specify the amount of the host system’s processor you want to allow a given virtual machine to consume. By clicking the Resource Allocation node in the Web interface, you can specify Relative Weight for each virtual server, giving one server CPU priority over another, or you can apply reserves and limits of CPU to each machine. Doing so enables you to prevent one virtual machine from consuming all the host system’s resources and then impacting the performance of other virtual machines. Figure 8.7 shows the Resource Allocation interface.

You can allocate CPU resources to virtual machines currently configured on Virtual Server. In most cases, you will only need to configure the Relative Weight of the virtual machines. You can also configure absolute limits. For more information, click the help button.

Virtual Machine ▲	Relative Weight	Reserved Capacity (% of one CPU)	Maximum Capacity (% of one CPU)	Reserved Capacity (% of system)	Maximum Capacity (% of system)	CPU Usage
Virtual Machines Currently Running						
No virtual machines are currently running.						
Total Capacity Reserved				0%		
Available Capacity Remaining				100%		
Virtual Machines Not Currently Running						
VSTerminal Server ▶	<input type="text" value="100"/>	<input type="text" value="0"/> %	<input type="text" value="100"/> %	0%		12.5%
VSTerminalServer2 ▶	<input type="text" value="75"/>	<input type="text" value="0"/> %	<input type="text" value="100"/> %	0%		12.5%

Figure 8.7: Allocating CPU resources to virtual machines.


Implementing Terminal Services in a Virtual Environment

To install WS2K3 and the terminal server role in a virtual environment, follow the same procedures outlined in Chapter 2. You can use manual installation, unattended installation, or Sysprep. If you want to use RIS, however, you will have to use a boot floppy, as Virtual Server does not support PXI boot as of this writing.

 VMWare does support PXI boot its products.

If you need to build a number of identical virtual machines, follow these steps:

1. Create your first virtual machine and install the OS, the Terminal Server role, and any applications that will be on all virtual machines.
2. Run Sysprep to reset the OS to boot mode, and shut down the virtual machine.
3. Before booting the virtual machine, create a copy of the virtual hard disk created by the first virtual machine. The copy can then be cloned to create as many virtual machines as you need with this configuration.
4. To mount a Sysprep-create virtual hard disk, you will select the Use Existing Virtual Disk option in the *Create new virtual machine* interface, and specify the copy you want to use.


 You might also want to maintain frequent backups of the virtual disk files, as they are, in effect, your entire server. These backups can be used to restore a virtual machine to a previous known-good state. This process is much easier and faster than restoring an entire physical disk from tape.

With Virtual Server, there is no special configuration needed in order to run Terminal Services within a virtual machine. Memory and processor utilization are managed by Virtual Server in the same manner for terminal servers as for any other type of virtual machine. VMWare, however, has a special setting specifically for Citrix Terminal Servers, which increases the initial allocation of memory to the virtual machine, making it faster to respond to new user session requests.


Why Virtualize?

At first look, it can be very tempting to virtualize your entire terminal server environment. The ease of backups, rapid deployment, and differential disks are all very appealing to maintain an efficient and highly available terminal server infrastructure. You must, however, weigh the cost of using virtualization against the benefits. Keep in mind that in a virtual environment, the whole is not always equal to the sum of its parts. You will not get the same performance out of four virtual machines as you would from four physical servers. There is the overhead of the host OS as well as the level of context switching that the physical CPU must perform to manage requests from multiple OSs.

What drew a lot of attention to VMWare and terminal server was that Win2K could only support 2GB of kernel memory. Thus, even if you built a Win2K terminal server with 16GB of RAM, it would only be able to support a few more concurrent users than a server with 4GB of RAM. Virtualization was an easy fix. Break the large physical server into smaller virtual machines, each using its maximum 2GB of kernel memory. WS2K3 handles kernel memory very differently and is therefore not impacted by the 2GB limit.

 Read your licensing agreements carefully when implementing virtual servers. In most cases, Microsoft included, you will need the same type of licenses for a virtual server as you would for a physical server. For example, if you are running four virtual machines on a Microsoft Virtual Server, you would need five OS licenses—one for the host and one for each of the virtual machines.


One rather unique method for combining virtualization and Terminal Services that has received a lot of press lately is to use Terminal Services to serve virtual workstation OSs to end users. This setup is in direct competition with the blade PC platform. Instead of purchasing blade PCs that your users RDP to, you instead use Virtual Server to host multiple virtual machines each running your desktop OS. Users then connect to their own virtual desktop via either a direct RDP connection (as they would with a blade) or via Citrix Presentation server (giving you the advantages of the ICA protocol in a desktop OS environment).

 Brian Madden has written an excellent series of articles on this subject (<http://www.brianmadden.com/subject.asp?ID=29>) and Citrix has posted instructions on how to publish virtual desktop OSs via Presentation Server (<http://support.citrix.com/article/CTX108077&searchID=15959050>).

Application Isolation

Another type of virtualization is performed at the application level. Depending on the third-party product you are looking at, this application-level virtualization can be referred to as either isolation or virtualization. (Throughout, this section will refer to this family of utilities as application isolation).

Application isolation enables you to run mutually incompatible applications on the same server without the overhead of an entire virtual machine. With application isolation, the application is installed into a virtual environment within the OS. Most of these products mirror the file structure of a default installation of Windows so that the isolated application can have its own copy of critical system DLLs even if they are older than the actual DLL running within the OS. These products also virtualize the registry so that one registry key can be set to different values for each application.

 Citrix Presentation Server 4.0 includes application Isolation technology. Softricity's SoftGrid takes application isolation even further by storing all files and registry settings for an isolated application within a single, streamable file, enabling you to virtually install applications immediately upon launch by an end user.

Summary

Maximizing server utilization while maintaining acceptable end user performance is the primary objective of most infrastructure projects. With native tools such as WSRM and third-party options, you can strive to make your terminal servers as efficient as possible by reducing runaway and leaky applications.

Once your servers are running at peak efficiency, you can then look to virtualization technologies, if needed, to reduce wasted CPU cycles on your physical servers by combining multiple virtual systems onto one physical server. The “nirvana” of server efficiency would be an environment in which all your physical computers operated at 80 percent processor most of the time with no degradation in performance. Given the large number of variables in a terminal server environment—user applications, profiles, multiple user sessions, and work patterns—this is a very difficult goal to achieve, but it is still worth pursuing.

I hope you have enjoyed this guide and have learned a few things along the way. Terminal servers are playing a larger role within enterprises today, serving mission-critical applications and full desktop environments for users. A thorough understanding of terminal server technology and how to design, install, configure, and maintain Terminal Services can serve you well in your role as a Windows systems administrator.

Download Additional eBooks from Realtime Nexus!

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

Appendix A: Terminal Services Clients

Remote Desktop Connection for Windows Server 2003

<http://www.microsoft.com/downloads/details.aspx?FamilyID=a8255ffc-4b4a-40e7-a706-cde7e9b57e79&DisplayLang=en>

☞ If you need the client in MSI format (for distribution via Group Policy) use
`msrdpcli.exe /c`
from a command prompt.

Remote Desktop Connection Client for Mac

<http://www.microsoft.com/downloads/details.aspx?FamilyID=c669fcf7-c868-4d45-95f3-f75ddd969232&DisplayLang=en>

Remote Desktop Web Connection

<http://www.microsoft.com/downloads/details.aspx?FamilyID=e2ff8fb5-97ff-47bc-bacc-92283b52b310&DisplayLang=en>

Terminal Services Client for PocketPC (2000 and 2002)

<http://www.microsoft.com/windowsmobile/resources/downloads/pocketpc/tsc.msp>

Appendix B: Important URLs

Windows Server 2003 Terminal Services Technology Center

<http://www.microsoft.com/windowsserver2003/technologies/terminalservices/default.mspx>

Microsoft TechNet: Terminal Services

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/technologies/terminal/default.asp>

Terminal Server Sizing Sample Scripts

<http://www.microsoft.com/windows2000/techinfo/administration/terminal/loadscripts.asp>

Windows Server 2003 Terminal Server Licensing

<http://www.microsoft.com/windowsserver2003/techinfo/overview/termservlic.mspx>

Terminal Services Client Access License Activation Web site

<https://activate.microsoft.com>

Appendix C: Registry Changes

Listed here are all of the registry changes described in Chapter 2. I have formatted them to allow you to copy and paste them into a .REG file for easy import.

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Policies\Explorer]

"LinkResolveIgnoreLinkInfo"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server]

"IdleWinStationPoolCount"=dword:00000005

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp\UserOverride\Control Panel\Desktop]

"AutoEndTasks"="1"
"CursorBlinkRate"="-1"
"DragFullWindows"="0"
"MenuShowDelay"="10"
"WaitToKillAppTimeout"="20000"
"SmoothScroll"=dword:00000000
"Wallpaper"="(none)"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp\UserOverride\Control
Panel\Desktop\WindowMetrics]

"MinAnimate"="0"
```

Appendix D: Script Reference

USRLOGON.CMD

The following code is the modified USRLOGON.CMD that takes advantage of Win2K's ability to map the home directory to the user's folder. Deletions are in ~~STRIKETHROUGH~~ and additions are in **BOLD**. In order to use this script, you should set your ROOTDRIVE to the same drive letter that you use for your users' network home directories.

```
@Echo Off

Call "%SystemRoot%\Application Compatibility
Scripts\SetPaths.Cmd"

If "%_SETPATHS%" == "FAIL" Goto Done

Rem

Rem This is for those scripts that don't need the RootDrive.
Rem

If Not Exist "%SystemRoot%\System32\Usrlogn1.cmd" Goto cont0
Cd /d "%SystemRoot%\Application Compatibility Scripts\Logon"
Call "%SystemRoot%\System32\Usrlogn1.cmd"

:cont0

Rem

Rem Determine the user's home directory drive letter.  If this
isn't
Rem set, exit.
Rem

Cd /d %SystemRoot%\Application Compatibility Scripts"
Call RootDrv.Cmd
If "A%RootDrive%A" == "AA" End.Cmd

Rem
Rem Map the User's Home Directory to a Drive Letter
Rem
```

```
Rem
Rem Subst the user's profile directory onto the ROOTDRIVE
Rem if it is not already mapped as the Home Directory
Rem

if /I "%rootdrive%" == "%homedrive%" goto NoSubst

:DoSubst

Net Use %RootDrive% /D >NUL: 2>&1
Subst %RootDrive% "%HomeDrive%%HomePath%"
if ERRORLEVEL 1 goto SubstErr
goto AfterSubst
:SubstErr
Subst %RootDrive% /d >NUL: 2>&1
Subst %RootDrive% "%HomeDrive%%HomePath%"
:AfterSubst

:NoSubst

Rem
Rem Invoke each Application Script.  Application Scripts are
Rem automatically
Rem added to UsrLogn2.Cmd when the Installation script is run.
Rem

If Not Exist %SystemRoot%\System32\UsrLogn2.Cmd Goto Cont1

Cd Logon
Call %SystemRoot%\System32\UsrLogn2.Cmd

:Cont1

:Done
```

TSSHUTDOWN Wrapper

TSSHUTDOWN.EXE is the native utility used to shut down or reboot a terminal server. This utility warns users that the terminal server is about to shut down so that they have time to save their work. It then executes a logoff command within each session, and finally shuts down the server. Unfortunately, the default behavior of the utility is to shut down, and not reboot, the server in 30 seconds.

I wrote this wrapper to prevent accidental shutdowns. Copy the following code, and save it as a VBS file in the SYSTEM32 directory. To launch the wrapper, type the VBS file's name at a command prompt (I name the file TSSHUTDOWN.VBS):

```
' AUTHOR: Greyson Mitchem, The Definitive Guide to WS2K3 TS
'
' This VBS file is a safety wrapper for the native
' TSSHUTDOWN.EXE that shuts down or reboots a TS.
' This script will collect the parameters for
' the tsshutdown command.

set oShell=CreateObject("Wscript.Shell")
set oNet=CreateObject("Wscript.Network")

returnkey=msgbox("This script collects the parameters needed
to"&VBCRLF&"correctly shutdown or reboot a Terminal
Server."&VBCRLF&"You may abort the process at any time by hitting
CANCEL."&VBCRLF&VBCRLF&"Do you wish to continue?", VBOKCancel +
VBInforamtion, "TS Shutdown")

IF returnkey=VBCancel THEN Wscript.Quit(1)

Server=InputBox("Enter the name of the server you wish to
Reboot/Shudown:", "Server Name", oNet.ComputerName)

if trim(Server)="" then Wscript.Quit(1)

bRestart=MSGBox("Do you wish to have the server
reboot?"&VBCRLF&VBCRLF&"Clicking NO will PowerDown the system
without rebooting.", VBYesNoCancel + vbQuestion, "Reboot or
PowerDown")

Select Case bRestart
    Case vbYes
        sOption="/REBOOT"
    Case vbNo
        sOption="/POWERDOWN"
```

```
Case vbCancel
    Wscript.Quit(1)
End Select

Wait=InputBox("Please enter the number of seconds to give users
to finish working before forcibly logging them off:", "Wait
Time", "60")
if trim(Wait)=" " then Wscript.Quit(1)


Delay=InputBox("Please enter the number of seconds to wait after
all users have logged off before shutting down the system:",
"Wait Time", "30")
if trim(Delay)=" " then Wscript.Quit(1)

' All Parameters have been collected. Now we build the comand
line.

ShutdnCMD="tsshutdown.exe "&wait&" /server:"&Server&" "&sOption&"
/Delay:"&Delay&" /v"

oShell.RUN(ShutdnCMD)

Wscript.Quit(0)
```

 If you want to create a shortcut to the wrapper, make the target
wscript %systemroot%\system32\<scriptname.vbs>

Maintenance Reboot Script

The following code is an example of a script that you can use to perform a maintenance reboot on a terminal server. Your circumstances—number of users, roaming profile environment, applications installed, and so on—will determine the frequency that you need to schedule maintenance reboots. In order for this script to run correctly, you need two additional files in the directory with the script: the resource it utility Sleep.EXE and a text file named yes.txt with the letter Y as its contents:

```
REM
REM Sending a message to any currently logged-on users
REM warning them that a maintenance reboot will occur
REM in 10 minutes.
REM

change logon /disable

msg * Please save your work and log off. Maintenance reboot in 10
minutes.

REM Pausing for 5 minutes
sleep 300

REM 5 minute warning
msg * Save your work now and log off. Maintenance reboot in 5
minutes

REM Pausing for 5 minutes
sleep 300

REM 30 second warning
msg * Maintenance reboot in progress. You will be logged off in
30 seconds

REM Pausing for 30 seconds
sleep 30

REM Logging all users off
logoff rdp-tcp < yes.txt
```

```

REM Stopping the Print Spooler service and deleting any
REM orphaned files.
net stop spooler
del %systemroot%\system32\spool\printers\*. * /q

REM Rebooting the TS
tsshutdn /REBOOT

```

Remote Desktop Web Connection Experience Dropdown

The Remote Desktop Web Connection client supports the same “Experience” features as the Remote Desktop Client. To add an experience dropdown to the native TSWeb page, make the following edits:

After the “Device redirection options” section (lines 238 – 242) insert:

```

'Experience Settings
select case document.all.comboExperience.value
case "1"
    MsRdpClient.AdvancedSettings2.PerformanceFlags = 0
case "2"
    MsRdpClient.AdvancedSettings2.PerformanceFlags = 7
case "3"
    MsRdpClient.AdvancedSettings2.PerformanceFlags = 7
case "4"
    MsRdpClient.AdvancedSettings2.PerformanceFlags = 15
case "5"
    MsRdpClient.AdvancedSettings2.PerformanceFlags = 111
end select

```

Change:

```

<!-- Column 1 spans 4 rows -->
<td valign="top" width="50%" rowspan=4>

```

To:

```

<!-- Column 1 spans 5 rows -->
<td valign="top" width="50%" rowspan=5>

```


Change:


```
<!-- Column 2 spans 4 rows-->
    <td rowspan=4 valign="top" align="left">
```

To:

```
<!-- Column 2 spans 5 rows-->
    <td rowspan=5 valign="top" align="left">
```

And finally, insert the following after the Window Size dropdown section (directly above “<!-- Row 3 -->”):

```
<tr>
<td valign="middle">
<p align="right"><label for="comboRes">Experience:</label></td>
<td valign="bottom">&nbsp;   <select size="1"
name="comboExperience" id=select class="topspace">
    <option value="3" selected><id
id="option15">Broadband</id></option>
    <option value="4"><id id="option16">56k
Modem</id></option>
    <option value="5"><id id="option17">28.8k
Modem</id></option>
    </select>
</td>
</tr>
```

 For additional information about the PerformanceFlags method, see:
http://msdn.microsoft.com/library/en-us/termserv/termserv/IMsRdpClientadvancedsettings_other_properties.asp

Appendix E: Terminal Services Command Line Reference

Change User

To toggle the system between Install and Execute Modes, use the following commands. Switches the terminal server into Install Mode:

```
CHANGE USER /install
```

Switches the terminal server into Execute Mode:

```
CHANGE USER /execute
```

Change Logon

Enables or disables new logons to the terminal server; does not affect currently logged on users.

To enable new logons:

```
CHANGE LOGON /enable
```

To disable new logons:

```
CHANGE LOGON /disable
```



When you reboot a terminal server, logons are automatically enabled (even if they were disabled when you shut down the system).

Query Terminal Servers

Lists all active terminal servers in the current or specified domain:

```
QUERY TERMSERVER [servername] [/domain:domain] [/address]  
[/continue]
```

where:

- *servername* is the name of a specific terminal server that you want to query
- /domain:*domain* is the name of the domain you want to query (the default is to query the current domain)
- /address includes the IP address of each server in the output
- /continue does not pause between screens of output

Query Session

Lists all current sessions on a specific terminal server:

```
QUERY SESSION [sessionname | username | sessionid]
[/server:servername] [/mode] [/flow] [/connect] [/counter]
```

where:

- *sessionname* is the name of a specific session that you want to query
- *username* is the name of the specific user you want to query
- *sessionid* is the ID of the specific session you want to query
- */server:servername* is the name of the server you are querying—the default is the server you are logged on to
- */mode* outputs the current line settings
- */flow* outputs the current flow control settings
- */connect* outputs the current connection settings
- */counter* outputs the counter information for the server

Query User

Lists all current users with sessions on a terminal server:

```
QUERY USER [username | sessionname | sessionid]
[/server:servername]
```

where:

- *sessionname* is the name of a specific session that you want to query
- *username* is the name of the specific user you want to query
- *sessionid* is the ID of the specific session you want to query
- */server:servername* is the name of the server you are querying—the default is the server you are logged on to



You can shorthand QUERY USER to QUSER

Query Process

Lists processes running on the terminal server (can be filtered to a specific session):

```
QUERY PROCESS [* | processid / username | sessionname | /id:nn |
programname] [/server:servername] [/system]
```

where:

- * lists all processes on the terminal server
- *processid* lists information about only the specific process ID
- *username* lists processes running under the context of a specific user
- *sessionname* lists processes running under the context of a specific session
- /ID:*nn* lists processes running in the session with the specified session ID number
- *programname* lists all processes started by the specified executable
- /server:*servername* is the name of the server you are querying—the default is the server you are logged on to
- /system lists processes running under the system context

Logoff

Logs a user out of the terminal server and deletes the session. If no arguments are included, the command logs you out:

```
LOGOFF [sessionid | sessionname] [/server:servername] [/v]
```

where:

- *sessionid* is the ID of the session you want to logoff
- *sessionname* is the name of the session you want to logoff
- /server:*servername* specifies the name of server on which the session you want to logoff is running (the default is the server to which you are connected)
- /v displays verbose information about actions being performed

Message

Sends a popup message to a user or users on the terminal server:

```
MSG [username | sessionname | sessionid | @filename | *]
  [/server:servername] [/time:seconds] [/v] [/w] message
```

where:

- *username* is the name of the user to whom you are sending the message
- *sessionname* is the session name to which you want to send the message
- *sessionid* is the ID number of the session to which you want to send the message
- *@filename* is the name of a text file containing usernames, sessionnames, or session IDs to which you want to send the message
- * sends the message to all users on the current or specified server
- /server:*servername* specifies the server where recipients of the message are connected
- /time:*seconds* the number of seconds to display the message before the popup closes itself
- /v displays information about the message as it is sent
- /w causes the popup window to wait for the user to click OK before closing
- *message* is the text of the message to send

Reset Session

Terminates a user's session without warning the user, and without performing a graceful logoff; can be used to terminate a hung session:

```
RESET SESSION [sessionname | sessionid] [/server:servername] [/v]
```

where:

- *sessionname* is the name of a specific session that you want to reset
- *sessionid* is the ID of the specific session you want to reset
- /server:*servername* is the name of the server on which the session resides
- /v displays verbose information about actions being performed

Shadow

Begins a remote control session:

```
SHADOW [sessionname | sessionid] [/server:servername] [/v]
```

where:

- *sessionname* is the name of a specific session that you want to shadow
- *sessionid* is the ID of the specific session you want to shadow
- */server:servername* is the name of the server on which the session resides
- */v* displays verbose information about actions being performed

Terminal Services Profile

Populates the Terminal Services profile path of a specified user; can also be used to copy the Terminal Services profile contents from one user to another (this command requires Administrator rights):

```
TSPROF /update [/domian:domainname | /local] /profile:path  
username
```

```
TSPROF /copy [/domian:domainname | /local] [/profile:path]  
src_user dest_user
```

```
TSPROF /q [/domian:domainname | /local] username
```

where:

- */update* populates the user *domainname\username*'s Terminal Services profile path with *path*
- */copy* copies the Terminal Services profile from *src_user* to *dest_user* and, if specified, updates *dest_user*'s Terminal Services profile path with *path*
- */q* displays the Terminal Services Profile path for the specified user


Terminal Server Shutdown

Shuts down or reboots a terminal server in an orderly fashion, giving any current users the opportunity to save their work and logoff:

```
TSSHUTDOWN [wait_time] [/server:servername] [/reboot] [/powerdown]
[/delay:logoffdelay] [/v]
```

where:

- *wait_time* is the number of seconds to wait after notifying the users that the terminal server is about to shut down before forcibly logging them off (the default is 30 seconds)
- */server:servername* is the name of the server to reboot/shutdown (the default is the server to which you are connected)
- */reboot* reboots the server
- */powerdown* powers down the server after Windows has shutdown; the server's BIOS must support this command
- */delay:logoffdelay* the number of seconds to wait after logging out all users before shutting down the system (the default is 30 seconds)
- */v* displays verbose information about actions being performed

 See Appendix D for a VBS wrapper for TSSHUTDOWN to make entering the parameters easier.

Remote Desktop Client Command Line Parameters:

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

where:

- *<Connection File>* specifies the RDP file for the connection
- */v:<server[:port]>* specifies the server name or IP address to connect to and the port on which to connect
- */console* connects to the console session of a WS2K3 system
- */f[fullscreen]* starts the client in full-screen mode
- */w:<width> /h:<height>* specifies a height and width for the connection window
- */edit* opens an RDP file for editing

/Migrate migrates legacy Client Connection Manager connections out of the registry and into RDP files.