How Terminal Services Works

In this section

Terminal Services provides the ability to host multiple, simultaneous client sessions on Microsoft® Windows® Server 2003. Terminal Server is capable of directly hosting compatible multi-user client desktops running on a variety of Windows-based and non Windows-based hardware. Standard Windows-based applications do not need modification to run on the Terminal Server, and all standard Windows Server 2003-? based management infrastructure and technologies can be used to manage the client desktops.

Back to Top

Terminal Services Architecture

Terminal Services consists of four components: the Windows Server 2003 multi-user kernel, the Remote Desktop client, the Terminal Services Licensing service, and Session Directory Services. Specifically:

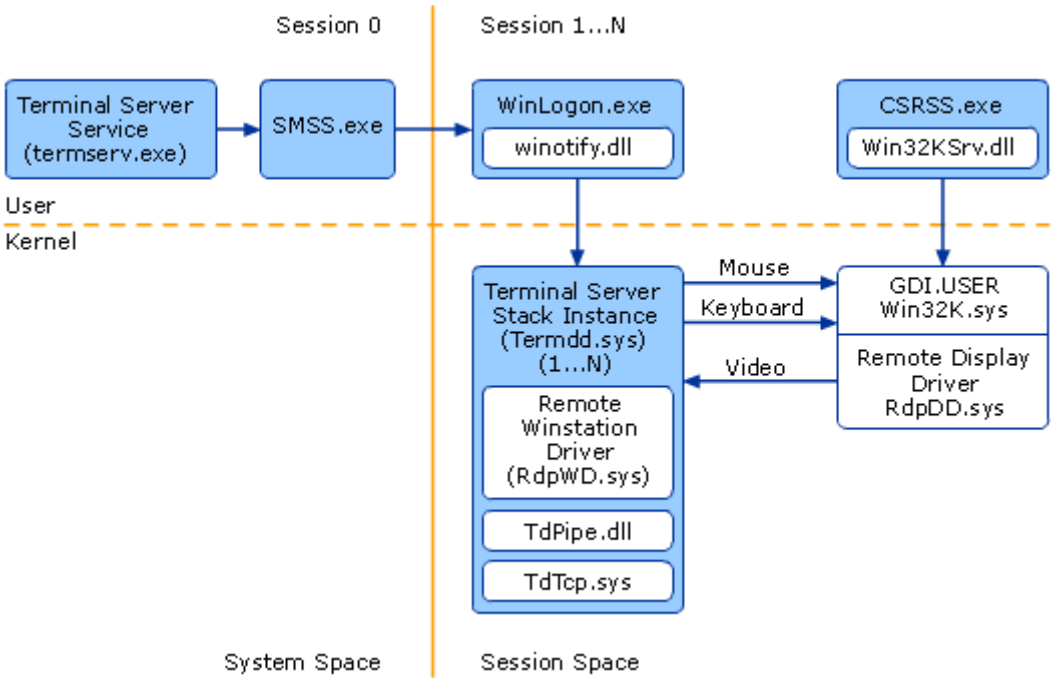
Multi-user kernel. The multi-user kernel extensions, originally developed for Windows NT 4.0 Server, Terminal Server Edition, have been enhanced and fully integrated as a standard part of the Windows Server 2003 family kernel. These are resident on the server at all times, regardless of whether Terminal Services is enabled or not.

Remote Desktop client: The client software is an application that establishes and maintains the connection between a client and a server computer running Terminal Services.

Terminal Services Licensing service: This system allows terminal servers to obtain and manage terminal server client access license (TS CAL) tokens for devices and users connecting to a terminal server.

Session Directory Services: The session directory (SD) keeps a list of sessions indexed by user name, and allows a user to reconnect to the terminal server where the users disconnected session resides and resume that session.

Terminal Services Architecture



The following table describes the Terminal Services architecture components.

Terminal Services Components

Component	Description
CSRSS.exe	The Client-Server Runtime Subsystem is the process and thread manager for all logon sessions.

RdpDD.sys	Captures the Windows user interface and translates it into a form that is readily converted by RDPWD into the RDP protocol
RdpWD.sys	Unwraps the multi-channel data and then transfers it to the appropriate session.
SMSS.exe	Session Manager creates and manages all sessions.
Termsrv.exe	Manages client connections and initiates creation and shutdown of connection contexts.
Termdd.sys	The RDP protocol, which listens for RDP client connections on a TCP port.
Tdtcp.sys	Packages the RDP protocol onto the underlying network protocol, TCP/IP.
Wlnotify.dll	Runs in the sessions WinLogon process to create processes in the user session.
Win32k.sys	Manages the Windows GUI environment by taking the mouse and keyboard inputs and sending them to the appropriate application.
WinLogon.exe	This system service handles user logons and logoffs and processes the special Windows key combination Ctrl-Alt-Delete. WinLogon is responsible for starting the Windows shell (which is usually Windows Explorer).

Terminal Services Architecture

As the Windows Server 2003 Terminal Server boots and loads the core operating system, the Terminal Server service (termsrv.exe) is started and begins waiting for session connections. Each connection is given a unique session identifier or "SessionID" to represent an individual session to the Terminal Server, and each process created within a session is "tagged" with the associated SessionID to differentiate its namespace from any other session namespaces.

The console session (Terminal Server keyboard, mouse, and video) is always the first to load, is treated as a special-case client connection, and is assigned SessionID0. The console session starts as a normal Windows Server 2003 session, with the configured Windows display, mouse, and keyboard drivers loaded.

After creating the console session, the Terminal Server service then calls the Windows Session Manager (SMSS.EXE) to create two idle client sessions, which then await client connections. To create the idle sessions, the Session Manager starts the Client-Server Run-time Subsystem (CSRSS.EXE), and a new SessionID is assigned to that process. The CSRSS process also invokes the WinLogon process (WINLOGON.EXE) and the Windows Manager and GDI kernel module (Win32k.sys) under the newly associated SessionID.

The Windows image loader recognizes this Win32k.sys as a SessionSpace loadable image by a predefined bit set in the image header. It then relocates the code portion of the image into physical memory with pointers from the virtual kernel address space for that session if Win32k.sys has not already been loaded. By design, it always attaches to a previously loaded images code (Win32k.sys) if one already exists in memory (that is, from any active application or session). The data (or non-shared) section of this image is then allocated to the new session from a newly created SessionSpace pageable kernel memory section.

Unlike the console session, Terminal Server client sessions are configured to load separate drivers for the display, keyboard, and mouse. The display driver is the Remote Desktop Protocol (RDP) display device driver (rdpdd.dll), and the mouse and keyboard drivers are replaced with the RDP driver Rdpwd.sys. These drivers allow the RDP client session to be both available and interactive, remotely. Finally, Terminal Server also invokes a connection listener thread for the RDP protocol (Termdd.sys), which listens for RDP client connections on a TCP port.

At this point, the CSRSS process exists under its own SessionID namespace, with its data instantiated per process as necessary. Any processes created from within this SessionID will execute within the SessionSpace of the CSRSS process automatically. This prevents processes with different SessionIDs from accessing another sessions data.

Back to Top

Terminal Services Physical Structure

Terminal Services provides remote access to a Windows desktop through "thin client" software, allowing the client computer to serve as a terminal emulator. It provides an effective and reliable way to distribute Windows-based programs, providing a single point of installation with multiple users having access to the Windows Server 2003 operating system desktop, where they can run programs, save files, and use network resources as if they were sitting at that computer.

For computers running Windows Server 2003 operating systems, the Terminal Services client program (Remote Desktop Connection) is already installed. Windows Server 2003 operating systems also include Terminal Services Client software for computers running 16- and 32-bit operating systems.

A Terminal Services client can exist in a variety of forms. Thin-client hardware devices that run an embedded Windows-based operating system can run the Terminal Services client software to connect to a server computer running Terminal Services. Windows, Macintosh, or UNIX computers can run Terminal Services client

software to connect to a Terminal Services server to display Windows-based applications. This combination of Terminal Services clients provides access to Windows-based applications from virtually any operating system.

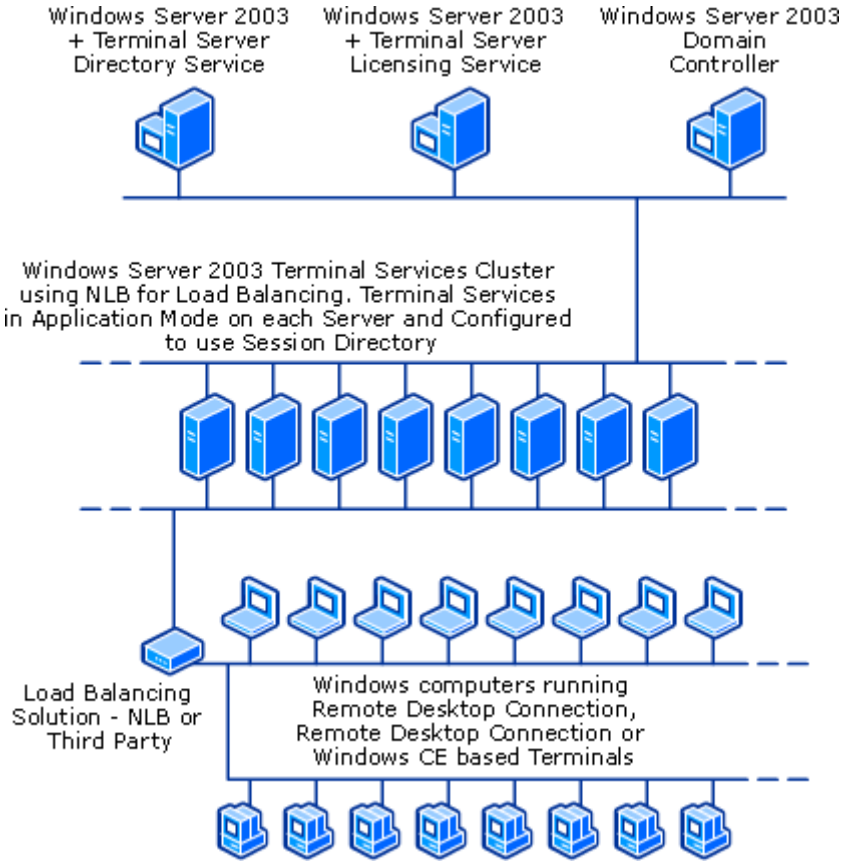
Terminal Server Licensing

The Windows Server 2003 operating system family provides a client license management system known as Terminal Server Licensing. This system allows terminal servers to obtain and manage terminal server client access license (TS CAL) tokens for devices and users connecting to a terminal server. Terminal Server Licensing is a component service of Windows Server 2003, Standard Edition; Windows Server 2003, Enterprise Edition; and Windows Server 2003, Datacenter Edition. It can manage unlicensed, temporarily licensed, and client-access licensed clients, and supports terminal servers that run Windows Server 2003 as well as the Windows 2000 Server operating system. This greatly simplifies the task of license management for the system administrator, while minimizing under- or over-purchasing of licenses for an organization. Terminal Server Licensing is used only with Terminal Server and not with Remote Desktop for Administration.

Session Directory

Terminal Services is a technology that lets users run Microsoft Windows-based applications on a remote Windows Server 2003-based computer. In a Terminal Server-based computing environment, all application execution and data processing occur on the server. In a load balanced environment, a farm of terminal servers have incoming session connections distributed in a balanced manner across the servers in the farm. The session directory (SD) keeps a list of sessions indexed by user name, and allows a user to reconnect to the terminal server where the users disconnected session resides and resume that session.

Terminal Services Physical Structure



Terminal Services Components

Component	Description
Terminal Server	Hosts applications for client computers.
Terminal Server Licensing Service	Issues TS Device CAL token to a requesting terminal server.
Session Directory	Maintains a list of the user names associated with the session IDs connected to the servers in a load balanced Terminal Server cluster.
Remote Desktop Connection	Client software that allows a connection to one remote computer.
Remote Desktops MMC Snap-in	Client software used by administrators to connect to multiple remote computers simultaneously.

Remote Desktop Web Connection	ActiveX RDP Web browser client.

Back to Top

Terminal Services Processes and Interactions

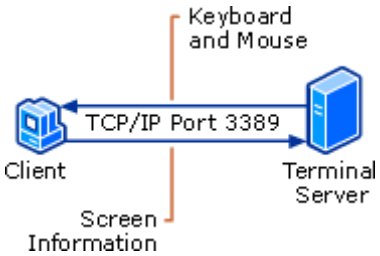
Terminal Services transmits only the user interface of the program to the client, with the client computer connecting through the network, sending keystroke and mouse-movement information over the Remote Desktop Protocol to the Terminal Server. It then sends the client screen information in the form of simple (and bandwidth-friendly) GDI events, backed with bitmap information if required to properly display the desktop state.

Each user logs on and sees only their individual session, which is managed transparently by the server operating system and is independent of any other client session. The Terminal Server provides virtual Windows session management, so users can essentially treat that session as their own personal computer.

Remote Desktop Clients

There are three clients for Terminal Services: Remote Desktop Connection, Remote Desktops Snap-In, and Remote Desktop Web Connection. The client software is a very small software application that establishes and maintains the connection between a client and a server computer running Terminal Services. Each client transmits all input from the user to the server, such as keystrokes and mouse movements, and all output from the server such as application display information and print streams. Remote Desktop Web Connection provides most of the same functionality as the Remote Desktop Connection software; but it does not require a private network, or virtual private network connection. Remote Desktop Web Connection is covered in the next section.

Connection using Remote Desktop Connection



Remote Desktop Connection and Remote Desktops Snap-In

The Terminal Services client has a new name, Remote Desktop Connection. Remote Desktop Connection can be installed and run on any Windows 95, Windows 98, Windows Millennium Edition, or Win32 platform (non-Windows-based clients are supported by the Citrix Metaframe add-on).

The client initiates a connection to the Terminal Server through TCP port 3389. The Terminal Server RDP listener thread detects the session request and creates a new RDP stack instance to handle the new session request. The listener thread hands over the incoming session to the new RDP stack instance and continues listening on the TCP port for further connection attempts. Each RDP stack is created as the client sessions are connected to handle negotiation of session configuration details.

First, an encryption level is established for the session. The Terminal Server initially supports three encryption levels: low, medium, and high.

Low encryption encrypts only packets being sent from the client to the Terminal Server. This "input only" encryption is to protect the input of sensitive data like a users password. Medium encryption encrypts outgoing packets from the client the same as low-level encryption, but also encrypts all display packets being returned to the client from the Terminal Server. This method of encryption secures sensitive data as it travels over the network to be displayed on a remote screen. Both low and medium encryption use the RC4 algorithm with a 40-bit key. High encryption encrypts packets in both directions, to and from the client, but uses the industry standard, non-exportable 128-bit high-level encryption.

At this point, prior to any logon being presented to the end user, the licensing details are negotiated. First, the client secures a Windows Server 2003 Client access license. Second, a Windows Server 2003 desktop license is verified on the machine that is connecting, and if no license can be validated, a connectivity license is provided to a non-Windows client to allow connection to the Terminal Server.

After session details have been negotiated, the server RDP stack instance, for this connection, is mapped to an existing idle Win32k user session, and the user is prompted with the Windows Server 2003 logon screen. If autologon is configured, the encrypted username and password is passed to the Terminal Server and logon proceeds. If no idle Win32k sessions currently exist, the Terminal Server service calls the Session Manager

(SMSS) to create a new user space for the new session. Much of the Win32k user session is using shared code and does load noticeably faster after one instance has previously loaded.

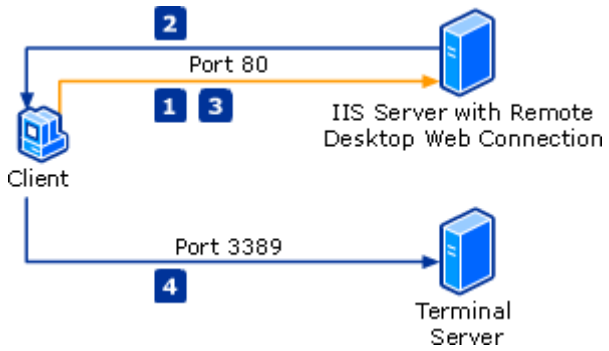
After the user types her or his username and password, encrypted packets are sent to the Terminal Server. The Winlogon process performs the necessary account authentication to ensure that the user has sufficient credentials to log on and then passes the users domain and username to the Terminal Server service, which maintains a domain/username, SessionID list. If a SessionID is already associated with this user, they are remapped into the existing namespace, and the previous session stack is reloaded. Otherwise, the connection proceeds as normal and the Terminal Server service creates a new domain/username, SessionID mapping. If for some reason more than one session is active for this user, the list of sessions is displayed and the user decides which one to make the reconnection.

The Remote Desktops snap-in is ideal for administrators who are remotely administering multiple servers or terminal servers. You can create Remote Desktop connections to multiple terminal servers or to computers running Windows 2000 Server or Windows Server 2003 family operating systems with the Remote Desktops MMC snap-in. A navigable tree display provides easy switching between connections.

Remote Desktop Web Connection

Remote Desktop Web Connection provides virtually the same functionality as Remote Desktop Connection, but delivers this functionality over the Web. When embedded in a Web page, Remote Desktop Web Connection can establish a Remote Desktop session with a remote computer, even if Remote Desktop Connection is not installed on the client computer. Remote Desktop Web Connection must be installed on a Web server with Internet Information Services (IIS) and Active Server Pages (ASP) enabled.

The Remote Desktop Web Connection is an ActiveX control that offers full feature parity with the standard Remote Desktop Connection client software offered in Windows XP and the Windows Server 2003 family of operating systems. The Remote Desktop Web Connection, like the Remote Desktop Connection which is installed by default, allows for connectivity to a Terminal Server. When installing the Remote Desktop Web connection on a specified computer the ActiveX control and sample ASP pages are installed. Internet Information Server is the delivery mechanism for the Remote Desktop Web Connection ActiveX control. This ActiveX control launches a Remote Desktop session to a Terminal Server computer from within Internet Explorer.



How Remote Desktop Web Connection works

Remote Desktop Web Connection connects to the Terminal Server as follows:

1. The user opens a Web browser and requests the initial Remote Desktop Web Connection (DHTML) login page.
2. The IIS server sends the page, and if this if the first time the client has connected, the user is also prompted to download the Remote Desktop ActiveX Control.
3. The user populates the connection information which includes the Terminal Server name.
4. The client computer creates a connection directly to the Terminal Server computer by using port 3389.

Connection Process

Remote Desktop Web Connection is a Web application that consists of an ActiveX control and DHTML pages. The DHTML pages provided only serve as a way to pass parameters to the Remote Desktop ActiveX control. Once the information is passed from the DHTML page to the ActiveX control, the Web page and the Web server no longer has a role in the process. Internet Information Server only serves as a mechanism for delivering the ActiveX control. When the user clicks **Connect**, the parameters from the DHTML page are passed to the control and a Remote Desktop connection to the Terminal Server is initiated.

When you connect to the IIS computer that is serving up the Remote Desktop Web Connection page, you are connecting over port 80. Upon connection to the Web page, the ActiveX control is downloaded to your client computer and stored in the default location for downloaded controls in Internet Explorer - %systemroot\Downloaded Program Files. From the supplied sample Web page, the name of the Terminal Server and the display resolution are passed as parameters to the ActiveX control. After these parameters are passed, the connect method on the control is called, and then a session is launched to the Terminal Server computer.

The Active X control on the client computer then creates a connection directly to the Terminal Server computer over TCP port 3389.

Note

- The Web client is the same as the full Remote Desktop Connection client without the entire configuration interface. It obtains these properties from the Remote Desktop Web Connection page, and not by any communication with the IIS computer itself.

Session Disconnect

If a user decides to disconnect the session, the processes and all virtual memory space remain and are paged off to the physical disk if memory is required for other processes. Because the Terminal Server keeps a mapping of domain/usernames and SessionIDs, when the same user reconnects, the existing session is loaded and made available again. An additional benefit of RDP is that of being able to change session screen resolutions, depending on what the user requests for the session. For example, lets say a user had previously connected to a Terminal Server session at 800 x 600 resolution and disconnected. If the user then moves to a different computer that only supports 640 x 480 resolution and reconnects to the existing session, the desktop is redrawn to support the new resolution.

Automatic Reconnection

Automatic Reconnection adds resilience to the Remote Desktop Connection client is Windows Server 2003. It is designed to recover from temporary connection losses due to network problems. Automatic Reconnection enables disconnected Terminal Services sessions to automatically re-authenticate to a Terminal Server without prompting the user for credentials.

Mobile users can greatly benefit from this feature. For example, with the previous versions of Terminal Server client, a user working with a wireless laptop which momentarily loses connectivity would be disconnected from the session and greeted with the message "The connection was ended because of a network error. Please try connecting to the remote computer again." Now with Automatic Reconnection, the user resumes the original session without needing to re-enter a password.

Reconnection Process:

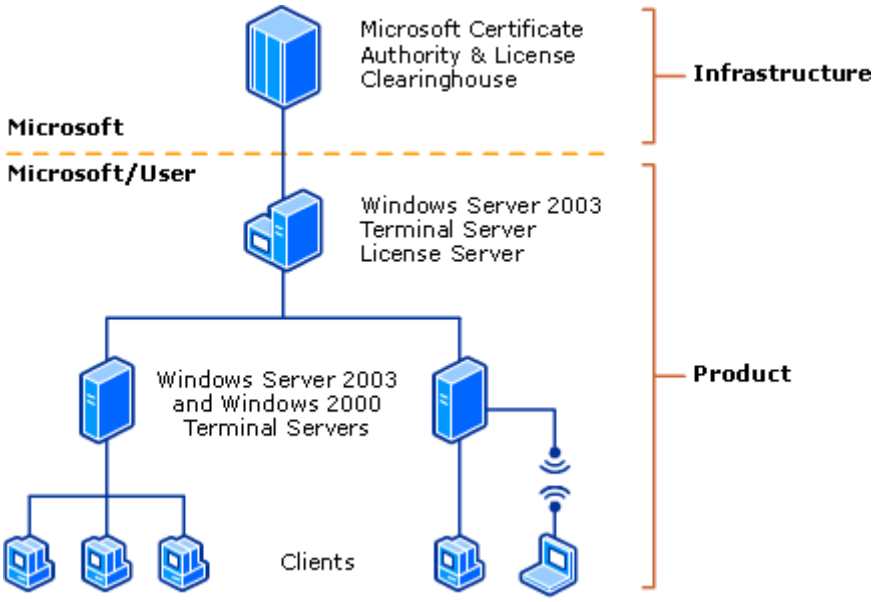
The Automatic Reconnection process is managed using a cookie that is sent to the client. When reconnection is initiated on the client, it sends this cookie to the server as a token for validating the connection. The auto reconnection cookie is generated at the server, and is flushed and regenerated any time a user logs in to a session or when a session is reset. This ensures that after the user connects to the session from a different computer, the original computer cannot reconnect. The server also invalidates and updates the cookie at hourly intervals, sending an updated cookie to the client as long as the session is active.

User Logoff

Once a user logs off from the session, all processes associated with the SessionID are terminated and any memory allocated to the session is released. Of course, if the user was running a 32-bit application like Microsoft Word and logged off from the session, the application would remain in memory until the very last user exited from the application.

Terminal Services Licensing service

The Windows Server 2003 operating system family provides a client license management system known as Terminal Server Licensing. This system allows terminal servers to obtain and manage terminal server client access license (TS CAL) tokens for devices and users connecting to a terminal server. Terminal Server Licensing is a component service of Windows Server 2003, Standard Edition; Windows Server 2003, Enterprise Edition; and Windows Server 2003, Datacenter Edition. It can manage unlicensed, temporarily licensed, and client-access licensed clients, and supports terminal servers that run Windows Server 2003 as well as the Windows 2000 Server operating system. This greatly simplifies the task of license management for the system administrator, while minimizing under- or over-purchasing of licenses for an organization. Terminal Server Licensing is used only with Terminal Server and not with Remote Desktop for Administration.



Terminal Server Licensing

Terminal Server for Windows Server 2003 (known as Application Server mode in Windows 2000 Server) provides application deployment and management for users on a variety of devices through its application server mode. Each device or user who initiates a session on a terminal server running Windows Server 2003 must be licensed with one of the following:

1. Windows Server 2003 Terminal Server Device Client Access License.
2. Windows Server 2003 Terminal Server User Client Access License.
3. Windows Server 2003 Terminal Server External Connector.

Note that additional licenses might be needed, such as Microsoft or other application, operating system, and Client Access licenses. The licenses in the preceding list are required even if other add-on products are used on top of Windows Server 2003.

The Terminal Services Licensing service is only associated with licensing for a terminal server client. It is not used to license any other application or service, and does not replace or interoperate with the licensing service for any other component, or alter your rights and obligations under any End User License Agreement (EULA). The Terminal Server Licensing service is not a replacement for purchasing a TS CAL.

TS CAL tokens are electronic representations of real licenses, but they are not actual licenses themselves. Therefore if a license token is lost, it does not mean that you have lost an actual license. If you have the documentation to prove that you have bought an actual license, the license token can be re-issued. Conversely, just because you have a license token does not mean that it necessarily maps to an actual legal license.

Terminal Services Licensing is designed to manage these license tokens to allow an administrator to more accurately assess an organizations licensing requirements. However, there are a few situations in which a license token will not map to an actual license. The administrator should determine if this is the case, and if necessary, purchase extra licenses (but not install the corresponding license tokens) to account for this discrepancy.

How Terminal Services Licensing Works

All communication during the licensing process occurs between the client and the terminal server, and between the terminal server and the license server. The terminal server client never communicates directly with the license server.

When a client device attempts to connect to a terminal server in Per Device mode, the terminal server determines if the client has a license token. Terminal server clients store license tokens in the following registry key:

HKEY_LOCAL_MACHINE\Software\Microsoft\MSLicensing

If a client has no license token, the terminal server attempts to contact a license server from its list of discovered license servers. If no contact is made, the terminal server restarts the discovery process. If no license server responds, the device can not connect to the terminal server unless it is operating within the terminal server grace period.

When a license server responds, the terminal server requests a temporary token for the device because this is the first time the device has connected to a terminal server. The terminal server then pushes this temporary token to the device. After a user has provided valid credentials resulting in a successful logon, the terminal server instructs the license server to mark the issued temporary token as validated.

The next time a user attempts to connect to a terminal server in Per Device mode from this device, the terminal server requests a Windows Server 2003 TS Device CAL token for this device. If the license server has available TS Device CAL tokens, the license server removes one token from the available pool, marks it as issued to the device, logs the device name, the user name of the device, and the date issued, and then pushes this TS Device CAL token to the device.

If the license server has no TS Device CAL tokens, it will first look to any other license server in its domain, workgroup, or site. License servers maintain information about where other accessible license servers exist, and if they have license tokens. If another license server is accessible that does have inventory, the first license server will request a license token from the second license server and deliver it to the terminal server, which then passes the token to the client device. If there are no available TS Device CAL tokens, the device will continue to connect with the temporary token.

Temporary tokens allow devices to connect for 90 days, and will then expire. TS Device CALs, while representing perpetual licenses, are set to expire 52-89 days from the date they are issued. The terminal server always attempts to renew these tokens 7 days prior to their expiration. The purpose of this system is to recover TS Device CAL tokens that are lost due to events such as hardware failure or operating system reinstallation.

Client License Distribution Per User

When a terminal server is configured in Per User mode, the terminal server must be able to locate a license server after the grace period has expired. While it is possible to install TS Per User CAL tokens on a license server, there is currently no method of assigning a TS Per User CAL token to a particular user account.

Client License Distribution for External Connector

There is currently no support in Terminal Server Licensing or the Microsoft Clearinghouse for the External Connector. In order to use an External Connector license, you will need to configure your terminal server in Per User mode.

Terminal Server Licensing Model

Terminal Server Licensing operates between several components as shown in the previous figure, including the Terminal Server Licensing-enabled license server, the Microsoft Certificate Authority and License Clearinghouse, one or more terminal servers, and terminal server clients. A single license server can support multiple terminal servers. There can be one or more license servers in a domain, or throughout a site.

Microsoft Certificate Authority and License Clearinghouse

The Microsoft Clearinghouse is the facility Microsoft maintains to activate license servers and to issue client license key packs to license servers. A client license key pack is a digital representation of a group of client access license tokens. The Microsoft Clearinghouse is accessed through the Terminal Services Licensing administrative tool. It can be reached directly over the Internet, through a Web page, or by phone.

License Server

A license server is a computer on which Terminal Server Licensing is installed. A license server stores all TS CALs license tokens that have been installed for a group of terminal servers and tracks the license tokens that have been issued. One license server can serve many terminal servers simultaneously. A terminal server must be able to connect to an activated license server in order for permanent license tokens to be issued to client devices. A license server that has been installed but not activated will only issue temporary license tokens.

Terminal Server

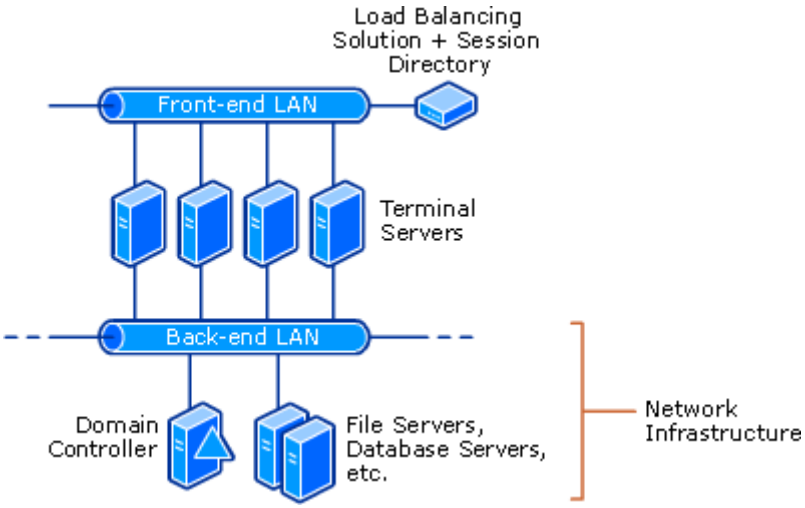
A terminal server is a computer on which the Terminal Server service is installed. It provides clients access to Windows-based applications running entirely on the server and supports multiple client sessions on the server. As clients connect to a terminal server, the terminal server determines if the client needs a license token, requests a license token from a license server, and then delivers that license token to the client.

Terminal Services Session Directory Service

In a Terminal Server-based computing environment, all application execution and data processing occur on the server computer. In a load balanced environment, terminal servers are grouped into farms, with each farm being represented to client machines as a single computer name with one IP address. The device performing the load balancing redirects incoming session connections to each machine in the farm according to its load balancing algorithm. Using a load-balancing solution with Terminal Server distributes sessions across the servers in the farm for improved performance. Terminal Services Session Directory, which is available with Windows Server 2003, Enterprise Edition, works with your load-balancing solution.

Session Directory is a load balancing feature that enables users to easily reconnect to a disconnected session on a server farm running Terminal Services. The Session Directory is a database that tracks user sessions that are running on load-balanced terminal servers. It provides information when a user reconnects (after disconnecting intentionally or because of a network failure) to ensure that the user reconnects to the same session rather than starting a new session. Session Directory, which can support several thousand sessions, is

also cluster-aware. Session Directory is compatible with the Windows Server 2003 load balancing service and is supported by third-party external load balancer products.



Terminal Services Session Directory management works with a load balancing service to ensure that users are transparently reconnected to the original server hosting their disconnected Terminal Server session.

The components of Terminal Services Session Directory are:

- A network load-balancing solution
- Two or more Terminal Servers logically grouped into a Terminal Server cluster
- A Session Directory server

Network Load Balancing, a clustering technology included in the Windows Server 2003, Enterprise Edition, and Windows Server 2003, Data Center Edition, enables servers to deliver high performance and failover protection. Network Load Balancing distributes IP traffic to multiple copies (or instances) of a TCP/IP service, such as Terminal Server or IIS, each running on an individual host within the cluster. The clients access the cluster using one or more virtual IP addresses; from there the NLB cluster evenly partitions the client requests among the hosts. From the clients point of view, the cluster appears to be a single server. As enterprise traffic needs increase, network administrators can simply add an additional server and incorporate it into the cluster.

In a load balanced environment, each Terminal Services cluster is represented to client machines as a single computer name with one IP address. The device or service performing its load balancing redirects connections to individual servers (nodes) in the cluster according to its load balancing algorithm.

Load balancing pools the processing resources of several servers using the TCP/IP networking protocol whereas Session Directory keeps track of the disconnected sessions on the cluster and ensures that users are reconnected to those same sessions.

Session Directory members can only be comprised of Enterprise and Datacenter Terminal Servers. Computers running Windows Server 2003, Standard Edition, are unable to join a Session Directory because they lack the required clustering technology; however a computer running Windows Server 2003, Standard Edition, can operate as the Session Directory server.

How Session Directory Works

1. An incoming connection to the cluster is load balanced to one node, which provides the logon prompt.
2. When the user logs on to the Terminal Server cluster, the Terminal Server receiving the initial client logon request sends the username to the Session Directory server.
3. The Session Directory server checks the username against its database and sends the result to the requesting server. The Session Directory database is a jet database containing a list of sessions indexed by username.
4. If the user has no disconnected sessions, the log on process continues at the server hosting the initial connection.
5. If the user has a disconnected session on another serer, the initial hosting server sends the client information necessary for the client to continue authentication against the server hosting the disconnected session. The transition from one server to the other is transparent to the user.
6. When the user logs on to the disconnected session, the Session Directory is updated.

The Terminal Server Session Directory database is updated and queried by the Terminal Servers whenever users log on, log off, or disconnect from a session.

Note

- The Session Directory redirection feature is dependent on the version of the Remote Desktop Connection

client used. The Remote Desktop client released with Windows XP is the minimum level necessary to function properly with Session Directory. Windows 2000 and Windows NT 4.0 Terminal Server clients are not Session Directory-aware, so client sessions will not be redirected to a disconnected session when that session is on an alternate node.

Network Load Balancing

Session Directory exists to solve a problem for Terminal Server clusters. These clusters can be load balanced using various solutions. This section discusses some general practices for any solution in use for load balancing. Here are some general considerations for Terminal Server clusters:

Consider splitting network traffic between two network adapters: one for Terminal Services connections and the other for access to other network resources and infrastructure. This allows for network access to the server in case the adapter bound to the cluster becomes unavailable.

For easier administration, place all load-balanced Terminal Servers into an Organizational Unit (OU) and apply Group Policy settings to that OU.

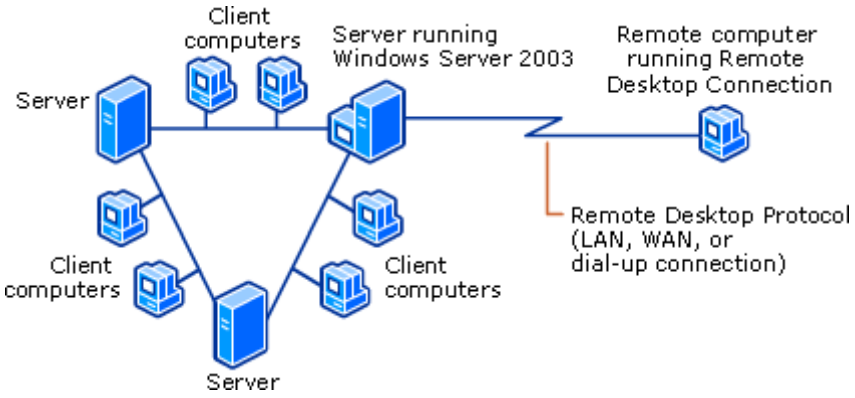
Home directories and other user data storage will need to be configured in such a way that the users can easily access their data no matter which server they are logged into.

Network Load Balancing (NLB) was previously installed by adding it as a service to a network connection and then configuring that network component on each node individually. NLB Manager in Windows Server 2003 is a new centralized snap-in added to provide easier configuration and maintenance for NLB configurations.

Although it is still possible to configure an NLB cluster by modifying network connection properties directly, the best practice is to use NLB Manager. In addition, the use of both NLB Manager and modifying network properties to configure an NLB cluster is not recommended.

Remote Desktop for Administration

You can use Remote Desktop for Administration to manage a network remotely using a configuration similar to the one shown in the following illustration.



Remote Desktop for Administration provides remote access to the server desktop by using the Terminal Services Remote Desktop Protocol (RDP) on port 3389. RDP transmits the user interface to the client session, and also transmits keyboard and mouse clicks from the client to the server. You can create up to two simultaneous remote connections. Each session you log on to is independent of other client sessions as well as the server console session. In essence, you can use Remote Desktop for Administration to log on to the server remotely as though you were logged on locally.

If you need to connect to the server console session remotely (for example, to access applications that direct only their user interface to the console), either use the Remote Desktops snap-in or use Remote Desktop Connection from the command line. When you attempt to connect to the console session, whether remotely or locally, you will be notified if there is already another user connected to the console session. The notification message will be shown after your logon credentials are validated, and will include information about the user who is logged on to the console session, including username, location of logon (local or remote), and the state of the session (in use, locked, or idle).

Important

- Be aware of the security implications of remote logons. Users who log on remotely can perform tasks as though they were sitting at the console. For this reason, you should ensure that the server is behind a firewall. You should require all users who make remote connections to use a strong password.

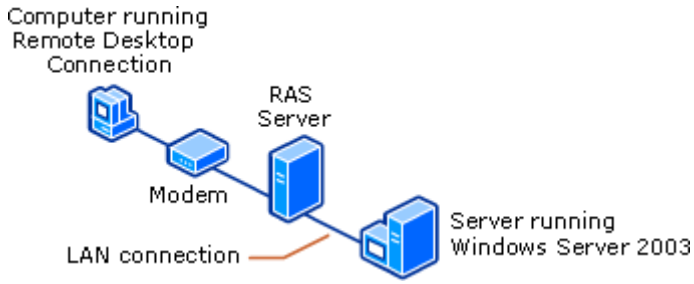
The connection to Remote Desktop for Administration uses TCP/IP, either over an existing network connection or by remote access. A remote access server running one of the Windows Server 2003 family of operating systems provides two different types of remote access connectivity:

Network and Dial-up Connections

Virtual private networking

The following illustration shows how you can connect to a computer running one of the Windows Server 2003

family of operating systems from a remote location using remote access.



Administering Windows Server 2003 family operating systems remotely

After you are connected to a computer running a Windows Server 2003 family operating system, you can use Remote Desktop for Administration to remotely administer the server and its local computers. Remote Desktop for Administration gives you access to a variety of administrative tools used to configure and manage computers. Through a Terminal Services session, you can access Microsoft Management Console (MMC), Active Directory, Systems Management Server, network configuration tools, and most other administrative tools.

Remote Desktop for Administration is extremely useful because it provides remote access to most configuration settings, including Control Panel, which usually cannot be configured remotely. Also, using Remote Desktop for Administration can be particularly convenient for diagnosing a problem and testing multiple solutions quickly.

You can access the servers from anywhere in the world by using a wide-area network (WAN), a virtual private network (VPN), or a dial-up connection. You can start time-consuming batch administrative jobs (for example, tape backups), disconnect, and later reconnect to the corporate network to check progress.

Server application and operating system upgrades can be completed remotely as well as tasks that are not usually possible unless you are sitting at the console, such as domain controller promotion/demotion and disk defragmentation. Server file system tasks such as copying large files and virus scanning are much more efficient when performed within a Remote Desktop for Administration session, rather than using utilities that are executed from a client computer.

Administrative tasks are quicker and more intuitive than using command line utilities, although it is still possible to open a command shell.

Note

- For some third-party applications, pop-up messages cannot be seen in a Terminal Services session. This is because there is a different security context or desktop for the connected session that does not display the applications pop-up messages. The pop-up messages in these instances will go directly to the console. If you need to see these messages, connect to the console session using Remote Desktop Connection from the command line or the Remote Desktops snap-in.

Back to Top

Network Ports Used by Terminal Services

Terminal Server uses RDP to communicate between client and server. RDP works only across a TCP/IP connection, such as a local area network (LAN), wide area network (WAN), dial-up, Integrated Services Digital Network (ISDN), digital subscriber line (DSL), or virtual private network (VPN) connection. You can still use other protocols, such as Internetwork Packet Exchange (IPX) or NetBIOS Extended User Interface (NetBEUI), as the transport protocol for non–Terminal Server traffic, such as network file or printer sharing, or between the client portion of a client-server application and its server.

Port Assignments for Terminal Services

Service Name	UDP	TCP
Remote Desktop Protocol (RDP)		3389

Back to Top

Related Information

The following resources contain additional information that is relevant to this section.

- What is Terminal Services?
- Terminal Services Tools and Settings

- For more information about Terminal Services, see MSDN.