

TERMINAL SERVICES: FROM A TO Z	2
ABOUT THIS GUIDE	2
INTRODUCTION	2
WHAT IS IT?	2
WHY TS?	4
GETTING STARTED	5
ENVIRONMENT	7
ACTIVE DIRECTORY PREPARATION	7
<i>OUs</i>	7
<i>Profiles/Home Directories</i>	8
<i>Groups</i>	12
TERMINAL SERVICES LICENSING	12
<i>Requirements</i>	13
<i>Licensing Modes</i>	14
<i>Licensing Server setup</i>	16
<i>Activating your licensing server</i>	17
<i>Adding licenses</i>	21
INSTALLING WINDOWS SERVER 2003	24
<i>Adding Terminal Services</i>	25
<i>Allowing User Access to your Terminal Server</i>	29
INSTALLING APPLICATIONS	30
<i>Basic Concepts</i>	30
<i>Application Control</i>	31
<i>Application Troubleshooting</i>	32
ACCESSING THE TS	35
<i>Full client</i>	36
<i>Web client</i>	37
<i>Other clients</i>	40
PRINTING	41
<i>Best practices</i>	42
<i>Alternatives</i>	42
TERMINAL SERVICES SECURITY	45
<i>Group Policies</i>	46
<i>Lockdown</i>	51
<i>Folder Redirection</i>	53
<i>Additional lockdown</i>	60
<i>External Access</i>	63
SCALING THE ENVIRONMENT	64
<i>Load Balancing</i>	65
<i>Scalability</i>	66
<i>Bandwidth Considerations</i>	67
ENHANCING THE ENVIRONMENT	68
<i>Seamless Windows</i>	68
<i>Firewall Friendly Access</i>	70
CONCLUSION	71

Terminal Services: From A to Z

About this Guide

After working with Microsoft Terminal Services (and pretty much all the add-ons available), answering thousands of questions on the subject at the Microsoft public newsgroups and Experts-Exchange (where I go by the tsmvp alias), myself and 2X decided to move ahead and publish this small, simple yet very valuable guide about Terminal Services.

As the name implies, the idea is to cover pretty much everything you need to know to properly deploy a terminal services based environment. Although there may be different ways to do certain things I will cover in this guide, everything here is based on my own experience in the field and all solutions described have been tested and in production for many years for many of my own customers. Again, they may not be the best or the ones by the book. But they do work and they are stable indeed.

Introduction

As you can see I do not assume any previous knowledge of Terminal Services; therefore we must explain a little bit about Terminal Services, how it works and why it may help you.

Before you guys go ahead and email me, bashing this guide, I just want to clarify a couple points. First of all this guide does not intend to be an in-depth book about Terminal Services and how to do everything related to it; secondly it is not meant to be 100% technical, written for people with years of experience with Terminal Services.

This guide is for everyone out there considering a Terminal Services deployment, but with no experience, or very little knowledge on the topic. It will lay down the foundations to successfully deploy a terminal services environment, giving you a solid understanding on how all this works which will help you immensely as you progress with your Terminal Services skills.

What is it?

Terminal Services, as it is today, is an old technology wrapped in some new, fancy wrapping paper.

If you are old enough, or if you have read at some stage how computing environments were back in the 60s, you probably heard the word 'Mainframe' and/or 'Dumb Terminals' (those almost like computers, with a green screen terminal and a keyboard). The idea behind them was quite simple: one big box (the mainframe) was responsible for running all the applications and processing all the data at a central location. In order to run applications, users would connect to the mainframe using the so-called 'dumb terminals'. These had no local processing power at all; they would simply send the keyboard entries back to the mainframe and the mainframe would send back the screen updates. So although users could 'see' their text based applications on their screen, everything was actually happening at the mainframe.

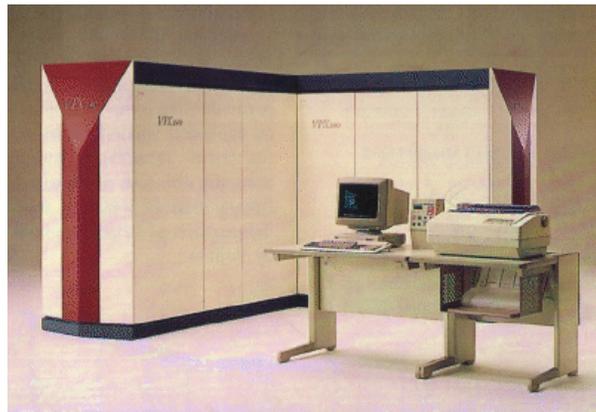


Fig 1
Mainframes and dumb terminals

Fast forward to today's environments and this whole 'centralized computing environment', sometimes referred to as 'Server Based Computing', is back in full swing but of course with a revamped interface. Exactly like in the 60s, today's server based computing environments centralizes all the applications and is responsible for all the required processing power. As you can see the main difference is simply the interface. Everything today relies on a GUI and a mouse so the old 'mainframe' idea just got updated to do the same old tricks but using today's interfaces. Terminal Services is simply a Windows Server based component (available on Windows 2000 Server and up) that delivers a unique 'desktop-like' environment to multiple users at the same time, all running off a single server (or multiple servers for high availability purposes). The same old tricks but with a couple updates.



Fig 2
Today's Server Based Computing GUI

Server based computing takes care of the processing required to run applications and the applications themselves, allowing users to access these resources from pretty much any device with little processing power and no applications installed locally. Terminal Services is just the 'Microsoft Windows' way of doing that, giving users the familiar look and feel they are used to.

Why TS?

The main question many people ask when they start researching about Terminal Services is 'Why TS?'. There are many reasons why a Terminal Services (when I say 'Terminal Services' I mean server based computing in general – that idea of a centralized location that runs your applications) solution is the way to go and we will cover some of these here.

As with any other technology or solution, it is not perfect and more than that, not recommended for everything. Based on my experience with it (over 13 years now!) I can definitely say I can find a reason (or a need) for Terminal Services on every single company out there. The key thing is to determine where TS would work well and actually help your company.

Advantages of using Terminal Services::

- **Centralized.** When using TS, applications are all installed on the terminal servers and not on every single PC in your company. This means if you have 5 terminal servers (using current hardware and well behaved applications, you can probably have 75 users per server, simultaneously) and you are deploying something like SAP using it, you have only 5 machines to upgrade when a new SAP release comes out, instead of upgrading 375 user PCs. Much easier to manage 5 boxes than 375.

- **Performance.** Many applications out there work just great when you use them on your LAN. With VPNs becoming more and more popular, users can now connect to the office and work from home. Once you try to run that application from your PC at home, over a much slower link (remember, at the office you are connected probably at 100MBits or 1GBit and at home you usually have a 2-7MBits high speed DSL or Cable), performance will probably suck. If you try the same application over Terminal Services it will be pretty much the same experience as if you were sitting at the office. Remember that TS sends you the screen updates only and not the whole data the application is actually using. That data is transferred between the TS itself and your application back end.
- **Extended lifecycle.** As your applications now run on the Terminal Servers, the client machine does not need to be upgraded often (as all the hard work is done on the TS side), greatly reducing your costs on hardware upgrades.

Of course not all applications will work well under Terminal Services. Some good examples are graphic intensive applications (AutoCAD, Google Earth, 3D Studio, etc), resource intensive ones (MathLab, etc) or applications that require some local hardware to be present (i.e. applications that must deal with USB peripherals that require drivers to be installed on the local PC).

But again, I am certain if you look around your company you will find a place for Terminal Services. What about that old application you were considering spending huge amounts of money to port to a web version? You can probably have it running on TS and provide access to it from anywhere in the world! No changes required!

The lesson to be learned here is simple: TS is not for everything and not for everyone. But it can be an excellent tool and problem solver for many companies out there. It is up to you to determine where it can help you.

Getting Started

Before we proceed with the 'hands-on' part of this guide I must clarify a couple things. First of all this guide is all based on Windows Server 2003 technologies (and not Windows Server 2008). Although this may not be the latest technology out there and knowing you may be asking yourself why learn about Terminal Services on Windows Server 2003 and not 2008, keep in mind that most, if not all information you will read here applies to a Windows Server 2008 TS deployment.

Secondly I assume you have some knowledge of Windows Server in general (Active Directory, Group Policies, etc). And finally, as mentioned before, you can do many of the things described here in different ways, all leading to the same results. This does not mean you are wrong and I am right or vice-versa. It just means there are many ways to perform different tasks so just use the one you feel more comfortable with.

Environment

This guide is based on a Windows Server 2003 environment and for this test environment all I used was two virtual machines (you can run these on VMWare Server or Microsoft Virtual Server – both free products) running Windows Server 2003 SP1 Standard (you can use SP2, R2, etc).

The first virtual machine was setup to be a Domain Controller (with the usual DNS, DHCP, etc) and we will not describe here how to setup a DC. This machine will also be used as a simple file server with some file shares that we will need for user home directories and profiles. The second box simply has Terminal Services added (and we will show you how to add the Terminal Server role to a Windows Server 2003 machine).

Active Directory Preparation

In most environments, when Terminal Services is deployed, settings that are applied to the regular computers (i.e. desktop wallpapers, themes, ability to shutdown the machine, etc) are usually not applied to the Terminal Servers. This means a different set of rules should exist to deal only with the user experience when connected to a terminal server and these rules may not be the same ones that apply to the user regular machine (i.e. his desktop on his desk).

This guide assumes you have a properly configured and working Windows Server 2003 Active Directory environment but as you will see many of the concepts/ideas discussed will apply to pretty much any Active Directory version (2000, 2003 or 2008).

We also assume you have rights to perform such changes on your Active Directory. In case you do not have rights, you will need to discuss your requirements with your administrator guys.

OUs

The first change is quite simple. You should create an Organizational Unit (OU) where all your Terminal Servers will stay. To keep things easy to understand we normally create an OU called 'Terminal Servers' and then move all the computer objects (your TSs) to this OU, as seen below.

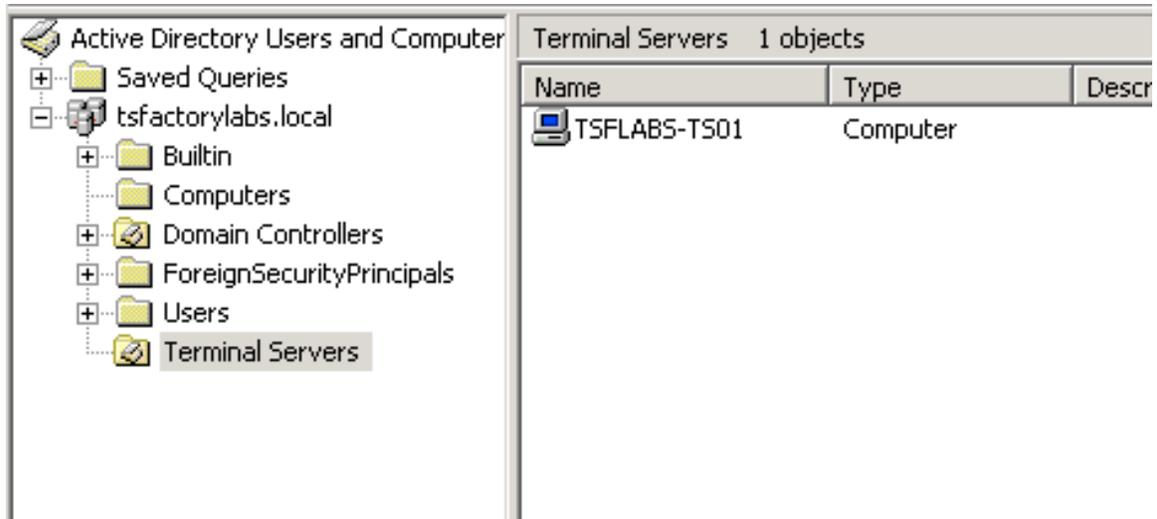


Fig 4
Active Directory Users and Computers, Terminal Servers OU

As you can see above I created the Terminal Servers OU (you can create it anywhere you want; in my case I created it at the root level) and moved my Terminal Server Computer Object (named TSFLABS-TS01) to it. To move computer objects simply right-click them and select 'Move'. Then browse to the OU you just created and click 'OK'.

Profiles/Home Directories

For a single server environment this step may not be required but I do recommend it as it will definitely help you when you decide to expand the environment by adding more terminal servers to serve your users!

Usually users logging in to a computer network will get assigned what we call a home directory. This is simply a unique location on the network where the user can save his files (i.e. Word documents, Excel spreadsheets, etc) and folders and most companies already have this set for its users so when they logon to their PCs a network drive is mapped to that location (i.e., an H: drive).

A profile in the other hand is a collection of user settings/preferences that are usually stored on the computer registry (some may be saved on files like MyApp.INI). When the user has a need to logon to multiple computers, the only way to make these preferences/settings follow the user is to save the profile to a network location that all computers can see. This is what we call a roaming profile.

As in our case users will be logging in on our terminal servers, we do not want the 'regular' profile (i.e., the one they use to save their preferences/settings on their Windows XP workstation) to be used for our terminal servers as these machines will not even run the same OS as the users' PCs! That is the reason

why under Active Directory Users and Computers, if you look at the properties for a user you will find a tab specifically for Terminal Services (Terminal Services Profile):

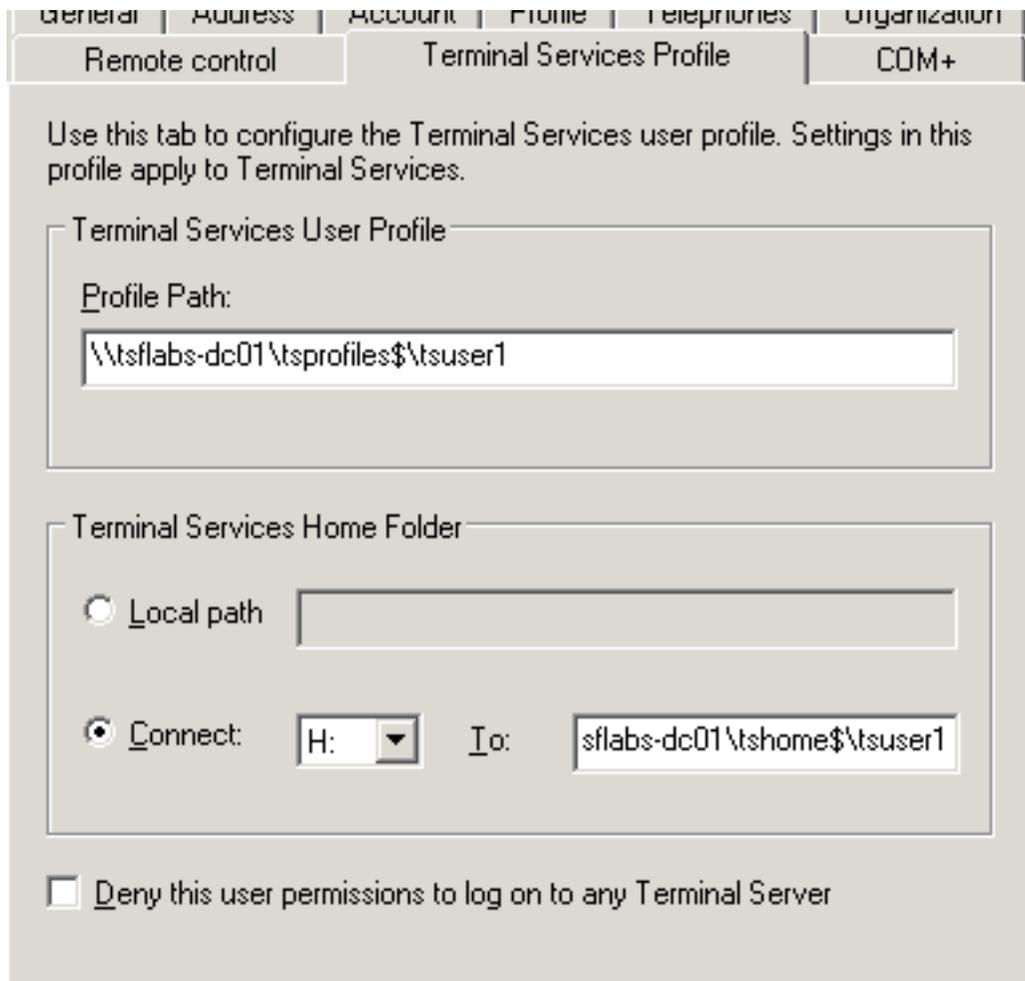


Fig 5
Active Directory Users and Computers, User Properties

What needs to be set here is the Terminal Services User Profile and the Terminal Services Home Folder. Usually the home folder is already set on the 'Profile' tab (right above the 'Terminal Services Profile' one). If you set it there, users will always get their home drive mapped, regardless whether the user logs in to a workstation or to a terminal server. And for this particular setting it is usually a great idea that your users always get the same home directory regardless of where they are logging in (so they can always find their files at the same place). So if it is already set under the 'Profile' tab there is no need to set it again under the 'Terminal Services Profile' tab.

A profile carries the user's settings and preferences. We may not want certain options on the user interface to be available when logging in to a terminal server.

In this case you must set a profile path under the 'Terminal Services Profile' tab. If you do not set it and you set one under the 'Profile' tab, it will be used for both regular workstations and TSs what we do NOT want. For this reason make sure you do set the 'Terminal Services User Profile' under the 'Terminal Services Profile' tab.

This setting will also allow your users to have the same settings/preferences regardless of the TS they are logging in to! Imagine having to setup Outlook options and preferences every single time you logon to a new TS. So later on, if you add a new terminal server to provide the same applications to your users, nothing else needs to be done regarding their profiles.

The steps you need to follow to setup the home directories and profiles are:

1. Create two folders on a file server (preferably not a TS as if that TS is down, all your shares will go down with it!) and share them with meaningful names (one folder for the home directories, assuming you do not have it already, and one for the terminal services profiles). In my example I created the folders TSHome and TSPProfiles and shared them as TSPProfiles\$ and TSHome\$. You must give your users enough rights to these folders and to the share (this usually means 'Read/Change' to the share and 'Read&Execute/Write/List Contents for the Folder itself – NTFS).

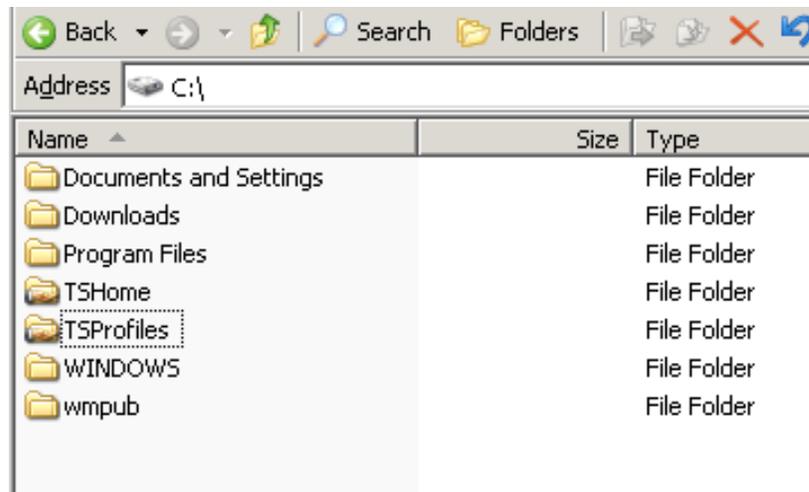


Fig 6
Folders for Home Directory and Profiles

2. Once the folders are shared, simply go to the user properties and set the 'Terminal Services Profile Path' (and if you did not set the Home Directory anywhere else, set it under 'Terminal Services Home Folder') to point to your file server TSPProfiles\$ share. In my case this will be [\\tsflabs-dc01\TSPProfiles\\$\%username%](\\tsflabs-dc01\TSPProfiles$\%username%). Note I am using the %username% variable so it gets resolved to the correct username once you press click

or Ok. For the Home Folder make sure you select 'Connect' and choose a drive letter that does not conflict with any other drive mappings you may have. And remember to point it to the TSHome\$ share (in my example, [\\tsflabs-dc01\TSHome\\$\%username%](#)).

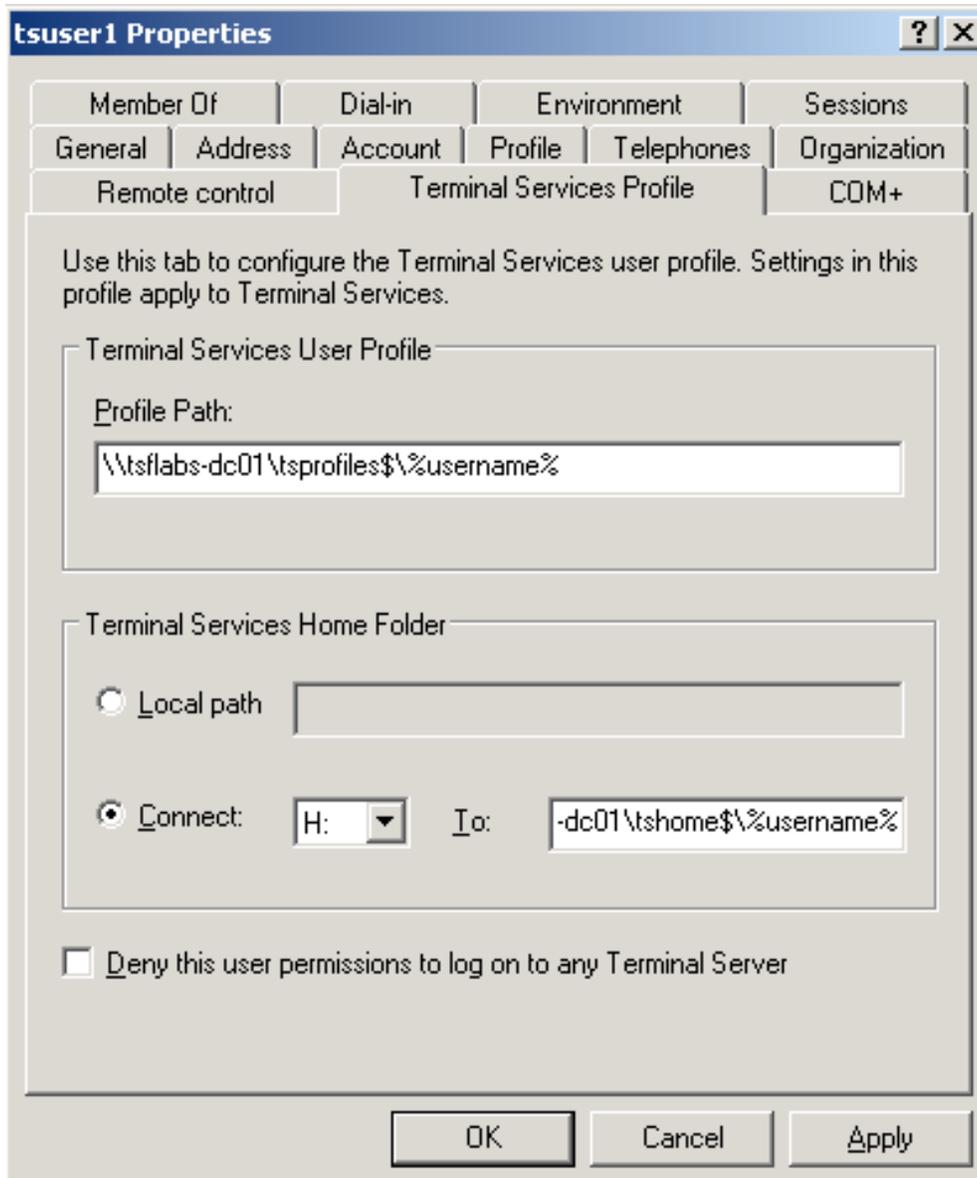


Fig 7

Active Directory Users and Computers, User Properties, Terminal Services Profile tab

3. Once you set all the above, the first time a user logs in you will see folders created under the TSHome\$ and the TSPfiles\$ for the user home directory and for his profile. For the profile, by default, administrators do NOT have rights to the folder. If you want administrators to have full rights over the user profile folder you should set this in a group policy (do not

worry about this right now; we will explain it in details later on this guide). The setting you must enable is shown below (under Computer Configuration | Administrative Templates | System | User Profiles. The setting is 'Add the Administrators security group to roaming user profiles).



Fig 8
Policy settings to give administrators rights on the user profiles

Note: As you get more familiar with group policies, there are many settings, including the TS Roaming Profile path that can be set using Group Policies and not necessarily in the User properties on AD!

Groups

For this particular environment we are setting up we will create a group called 'TS Users' and add all the users we want to provide access to the TS here and a group called 'TS Servers' and add all the TS computer objects to this group. Simply launch 'Active Directory Users and Computers' and create these two groups. Remember to add your users and your terminal servers to the respective groups we just created. We will need these down the road!

That is all for Active Directory!

Terminal Services Licensing

I am sure that one of the most discussed topics on the Microsoft Public Newsgroups or Experts-Exchange regarding Terminal Services is licensing. And I can definitely see the reasons for that. Since TSE was out (remember, TSE is the old Windows NT 4.0 Server, Terminal Server Edition) people did not understand exactly how licensing worked. Then with Windows 2000 Microsoft not only changed the licensing a little bit but also introduced licensing enforcement... To make things even more confusing different licensing options were introduced with Windows Server 2003 and old ones removed!

Yes, it seems confusing but if you read this section carefully I am sure you will understand how it works and will be able to figure out exactly what you need.

And before you ask me this question, Terminal Services is **NOT** a license saving solution or miracle; although you are installing applications on the server itself (i.e. Microsoft Office) what can mean a single machine (if you have a one server TS environment), this does NOT mean you will need only one single application license. Remember that multiple users will be able to access the application you just installed and therefore you must have as many licenses as needed to be legal. The savings you will have on a TS environment do NOT come from software licensing.

Another key thing to keep in mind is what your application EULA says about running it under Terminal Services. Certain applications may explicitly mention that running them under Terminal Services violates its EULA. So make sure you read the EULA for every single off-the-shelf application you intend to deploy under Terminal Services and in case of doubt, contact the manufacturer.

Requirements

For each client connecting to a Windows Server 2003 Terminal Server ("TS"), two licenses are required:

- **Windows Server, Client Access License (CAL).** A Windows Server 2003 Client Access License (CAL) is required for each user or device (or combination of both) that accesses or uses the server software. The same Windows Server 2003 Client Access License is used to access both Windows Server 2003 and Windows Server 2003 R2 servers.
- **Windows Server, Terminal Services Client Access License (TSCAL).** Terminal Server CALs are available in Per User/Per Device mode only. In Per User or Per Device mode, a separate TS CAL is required for each user or device that accesses or uses the server software on any server. You may reassign a TS CAL from one device to another device, or from one user to another user, provided the reassignment is made either (a) permanently away from the one device or user or (b) temporarily to accommodate the use of the TS CAL either by a loaner device, while a permanent device is out of service, or by a temporary worker, while a regular employee is absent. TS CALs are not available in Per Server mode as Windows sessions are not allowed in Per Server mode.

As you can see above, you must have two licenses in place for each user connecting to your TS: a CAL and a TSCAL. Usually the CAL is already in place in your company (as you need these to access any Windows Server you may have like a File Server, Print Server, etc) and are normally licensed per seat (although you can indeed license per server). Check with your network

administrator what licenses you have in your company and if they are per seat or per server CALs.

The TSCAL is needed by any device or user connecting to a terminal server, regardless of the OS they have on their machine (i.e. Windows 2000 Professional, Windows XP Professional, Windows Vista, Mac OS X, Linux, etc). If you are deploying a Windows 2000 Server Terminal Server then things are a little different regarding the TSCALs. If your clients are Windows 2000 Professional or Windows XP Professional, no TSCALs are required. But in the other hand there is no such thing as per user TSCAL... Well, sounds confusing? Let's explain the differences between Per User and Per Device TSCALs and why you should use one or the other.

Licensing Modes

With Windows Server 2003 Microsoft introduced Per User TSCALs and changed their policy regarding which Operating Systems (OSs) required a TSCAL. The difference between Per User and Per Device, and figuring out which one to use, is easy to understand. As you know everyone or everything connecting to a TS requires a license (TSCAL). The question you need to ask is if you have more users than devices or the other way around.

For example, assuming you have 50 users in your company but they may access the TS from their office computers (50, assuming you have one workstation per user), their own laptops, their friends PCs, Internet Kiosks and so on it is easy to see they will be accessing from multiple devices and the total number of devices at the end will be higher than the number of users. If that is your case Per User licensing is the way to go (as you will need less TSCALs).

In the other hand if you have only 25 computers in the office that your 50 users share during two different shifts and you do not provide access to the TS from anywhere else, it is clear you have fewer devices than users so it makes more sense (financially) to have Per Device TSCALs.

Another difference to consider is licensing enforcement. When the TS is set to use Per Device licensing, this is actually enforced. Once a user connects to a TS, a temporary TSCAL is issued, valid for 90 days. After this license expires the TS will try to get a permanent TSCAL from the licensing server and if such license is not available the connection to the TS will be denied. When set to Per User licensing, such enforcement does not happen and users will still be able to work (meaning it is up to you to make sure you have enough licenses to be legal).

As you can see the best way to handle licensing is to determine before hand what you have; more users or more devices. Once you know this, simply set the TS to use such licensing option (we will show you how to do it).

① There is also a special license called 'External Connector License'. This is basically an unlimited license to be used for non-employee access (i.e. general public, suppliers, partners, etc). If you are setting up your Terminal Services environment for non-employee access, this may be the way to go.

Licensing Server setup

Regardless of the licensing mode you choose, a Terminal Services Licensing Server must be available on your environment, otherwise users may not be able to logon to your TSs.

But before we go ahead and setup a licensing server, note that this will not happen immediately; once terminal services is installed you have 120 days to setup your licensing server and once this is done you have up to 90 days to add any licenses to it (what you will need for sure if you choose 'per device' licensing; Again, when selecting 'per user' licensing, of course you are required to have all the licenses you need to be legal but licensing is **NOT** enforced in this case).

i If you are not sure if TS is really the way to go in your particular case or if you need per device or per user licensing, use the 120 days grace period and wait to setup your licensing server; once that is done, you have another 90 days to determine the best licensing mode and if TS is indeed the way to go.

As this is a Windows Server 2003 environment your licensing server must be running on a Windows Server 2003 machine. If you still have Windows 2000 Server Terminal Servers still around, you can either use the new licensing server running on 2003 to handle Windows 2000 TSCALs (and of course the 2003 TSCALs) or simply keep the existing 2000 licensing server and setup a new one only for your 2003 Terminal Servers.

To install the licensing server on a Windows Server 2003 box follow these steps:

1. Logon to the machine as administrator and go to Control Panel | Add/Remove Programs | Windows Components and select 'Terminal Server Licensing' and click 'Next'.

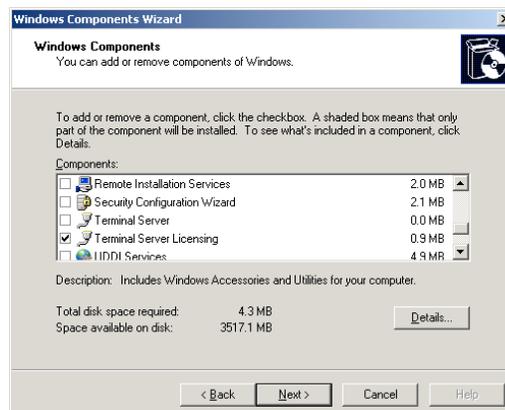


Fig. 9
Adding Terminal Server Licensing

2. Now you must decide if your licensing server (LS) will be an Enterprise License Server or a Domain License Server. The details on each are:
 - a. Enterprise License Server: first of all, an Enterprise LS cannot be installed on a stand-alone server; it must be installed on a domain controller or a member server in a domain. It is the right choice if your network has several domains and you want to maintain a single LS that will issue licenses to all TSs you may have on any domain.
 - b. Domain License Server: you can install a Domain License Server on a domain controller, a member server in a domain or a stand alone server. If you want to maintain a separate LS for each domain this is definitely the way to go.

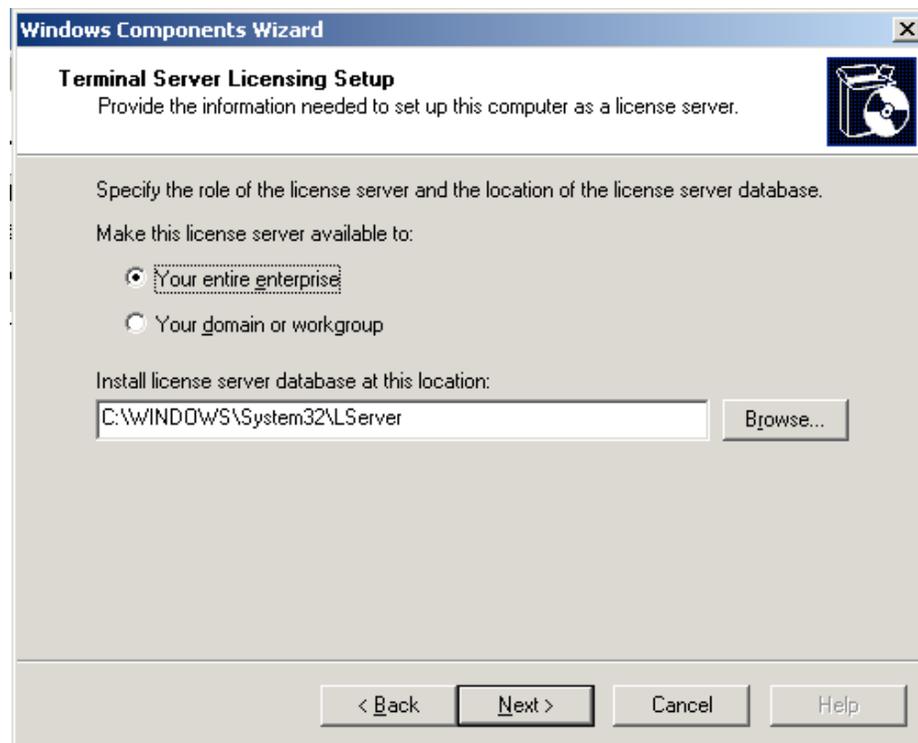


Fig. 10
Licensing Server Mode

Select the appropriate one for your environment and click 'Next'.
You just installed your first licensing server! Now let's activate it!

Activating your licensing server

Once your Licensing Server is installed and before adding any licenses we have an extra task to do: activate the Terminal Server Licensing server.

1. Click on Administrative Tools | Terminal Server Licensing. You should see your licensing server listed as 'Not Activated'.

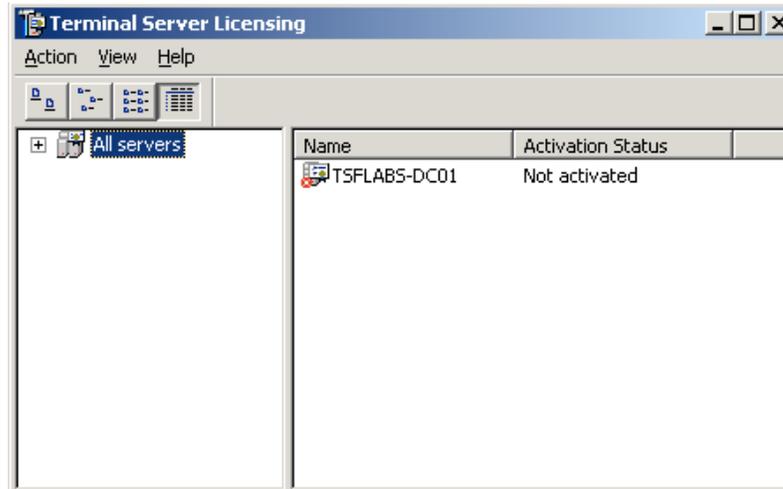


Fig. 11
Terminal Server Licensing

2. Right-click your LS and click 'Activate Server'.

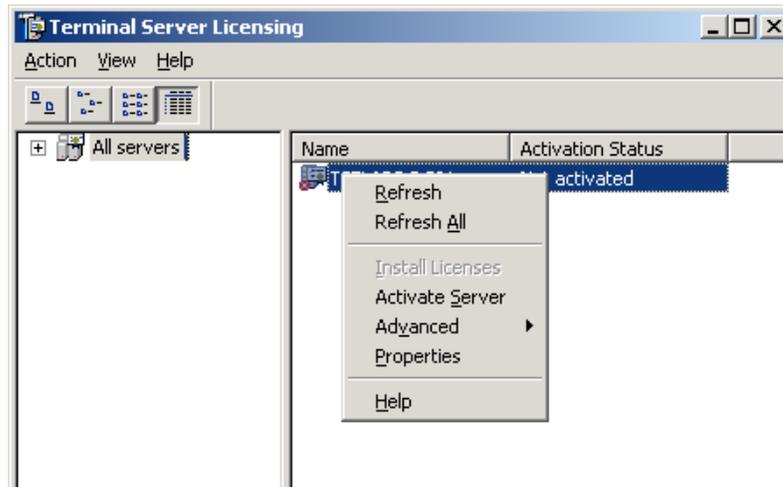


Fig. 12
Terminal Server Licensing Activation

3. The Activation Wizard screen will show up. Simply click 'Next'.

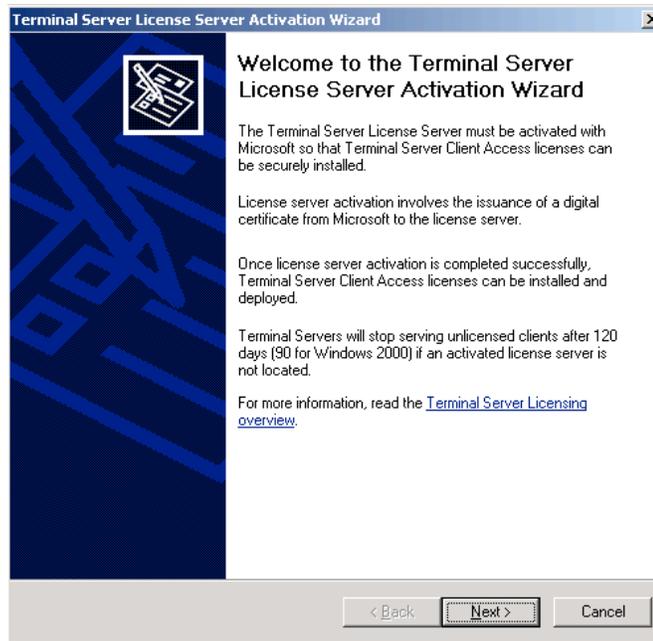


Fig. 13
Activation Wizard

4. Select the Activation Method and click 'Next'.

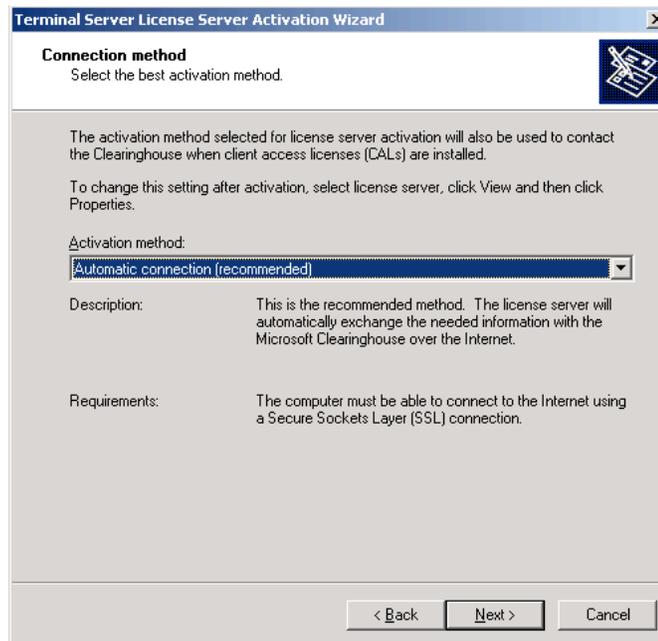
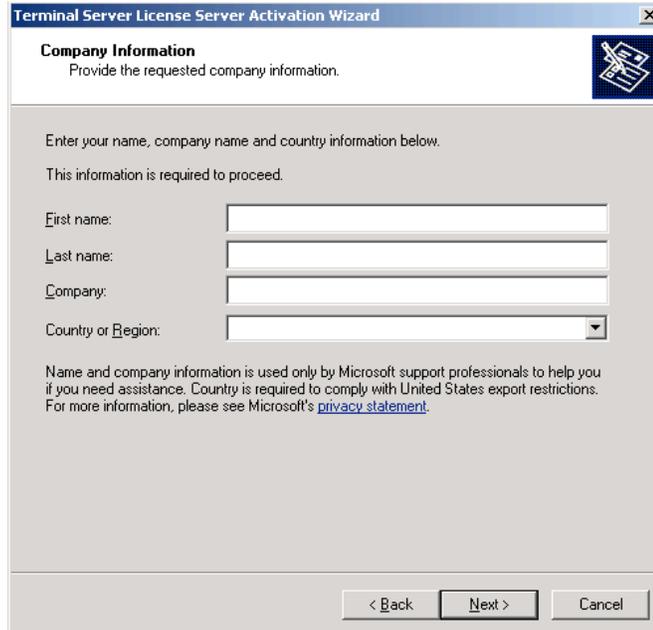


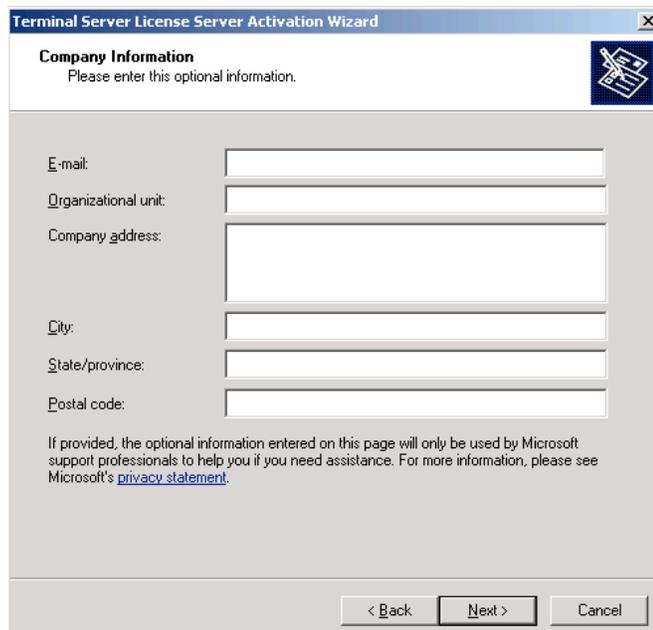
Fig. 14
Activation Method

5. In the following two screens, the Wizard will ask you some information (i.e. name, company, etc). Only the information on the first screen is mandatory. Type all that is required and click 'Next' on both screens.



The screenshot shows a window titled "Terminal Server License Server Activation Wizard" with a close button in the top right corner. The main heading is "Company Information" with the instruction "Provide the requested company information." and a small icon of a document with a pencil. Below this, the text reads: "Enter your name, company name and country information below. This information is required to proceed." There are four input fields: "First name:" (text box), "Last name:" (text box), "Company:" (text box), and "Country or Region:" (dropdown menu). At the bottom, there is a paragraph of text: "Name and company information is used only by Microsoft support professionals to help you if you need assistance. Country is required to comply with United States export restrictions. For more information, please see Microsoft's [privacy statement](#)." At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

Fig. 15
Activation Wizard - Mandatory Information



The screenshot shows a window titled "Terminal Server License Server Activation Wizard" with a close button in the top right corner. The main heading is "Company Information" with the instruction "Please enter this optional information." and a small icon of a document with a pencil. Below this, there are seven input fields: "E-mail:" (text box), "Organizational unit:" (text box), "Company address:" (text box), "City:" (text box), "State/province:" (text box), and "Postal code:" (text box). At the bottom, there is a paragraph of text: "If provided, the optional information entered on this page will only be used by Microsoft support professionals to help you if you need assistance. For more information, please see Microsoft's [privacy statement](#)." At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

Fig. 16
Activation Wizard – Optional Information

6. If you see the following screen you are all set! In case there are any problems, make sure you have an internet connection and that port 443 is not blocked to the outside.

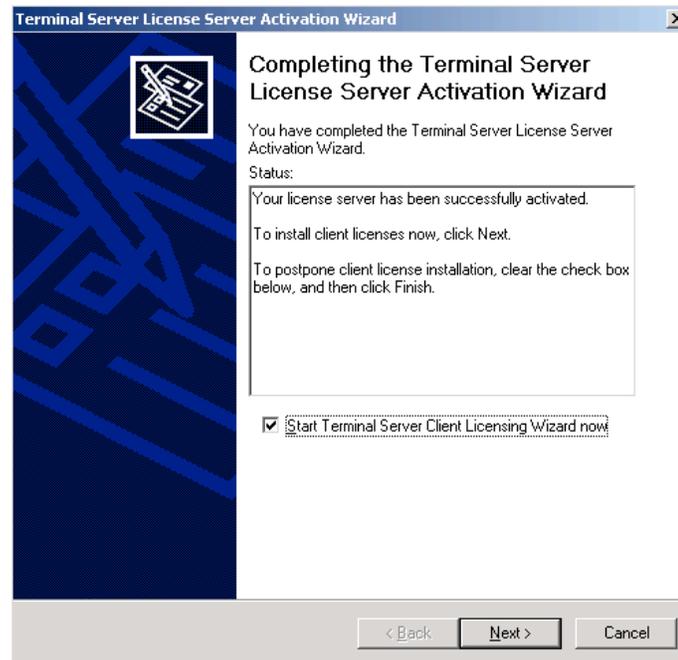


Fig. 17
Successful Activation

Adding licenses

The next step is to add licenses. Depending on how you get your licenses and on what agreement you may have in place with Microsoft, the actual licenses may differ. In certain cases it may be a 25 character code or simply an agreement number. Check with your company which agreement/licenses you have.

As an example, if you have a retail license pack, just follow these steps:

1. Launch Terminal Server Licensing. Your licensing server should be listed as 'Activated'.

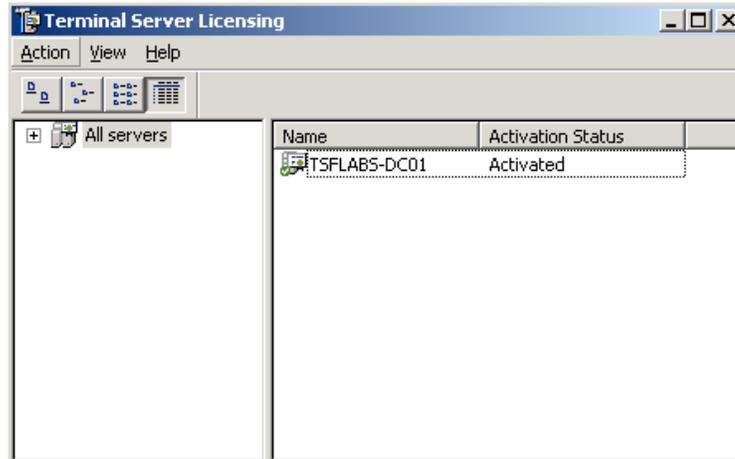


Fig. 18
Terminal Server Licensing

2. Right-click your LS and select 'Install Licenses'.

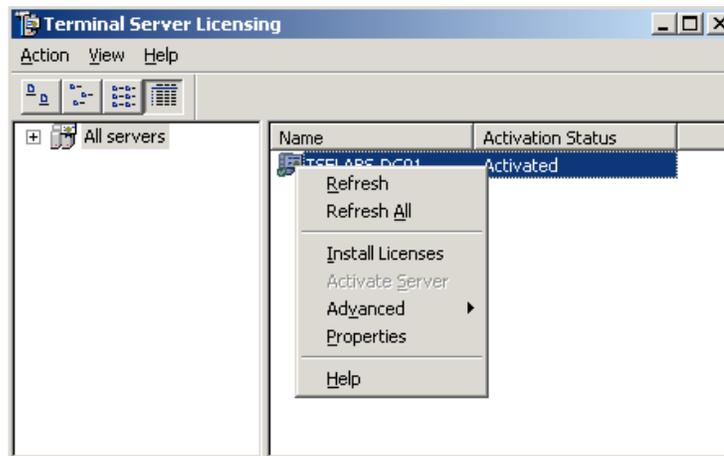


Fig. 19
Installing TSCALs

3. The TSCAL Installation Wizard will come up.

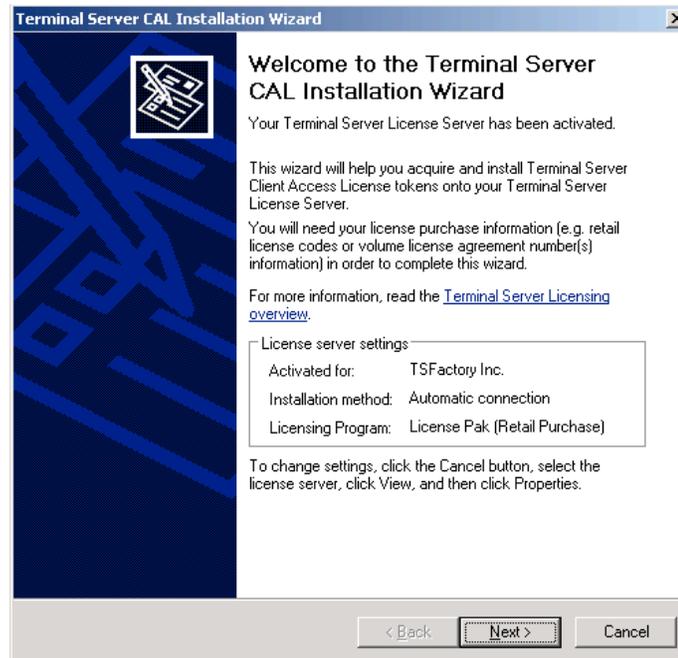


Fig. 20
Licensing Server Mode

4. Select your license program and click 'Next'.

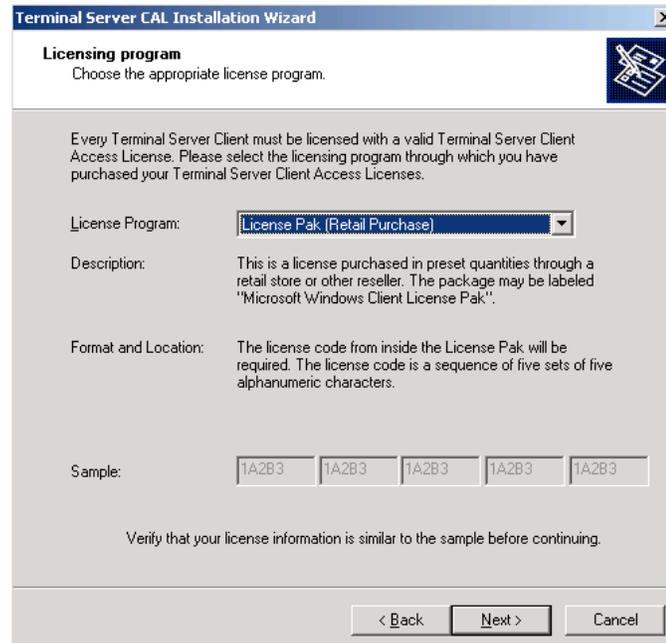


Fig. 21
Licensing Server Mode

5. Depending on the program selected the wizard will ask you the product code or license agreement number to proceed. Enter it and click 'Next'.

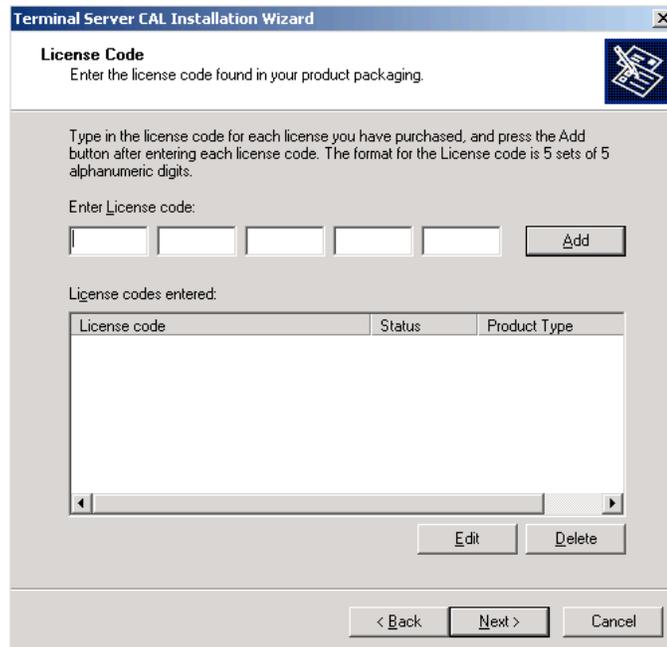


Fig. 22
Entering the license code

6. As explained before, depending on what you select on the first screen you may need to choose the licenses you want to install (i.e. Per Device TSCALs for Windows Server 2003) and then proceed to the final step. Once this is done your licenses will be shown on the Terminal Server Licensing window.

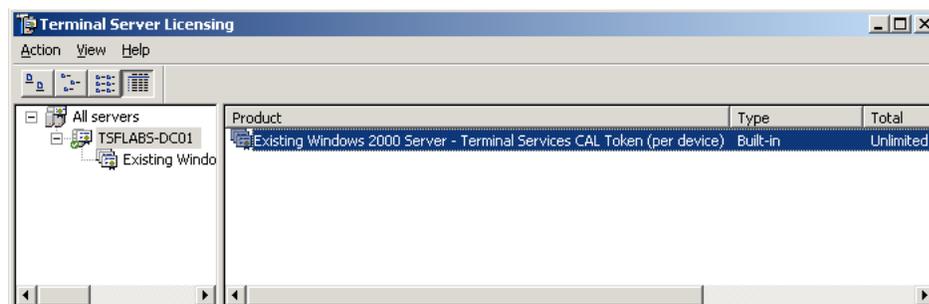


Fig. 23
Your license packs

Installing Windows Server 2003

Now that we have Active Directory all set for our users (well we will be dealing with Group Policies later but home directories, roaming profiles, etc are done for now) and an activated licensing server, the next logical step is to install our terminal server.

The installation itself is simply a standard Windows Server 2003 install; no additional options are selected and there are no hidden tricks at all! Trust me on that. Simply insert your Windows Server 2003 media on the server drive (or use the .ISO file with a virtual machine for example) and perform a plain, basic Windows Server installation. Once it finishes we will proceed with the terminal services installation.

Adding Terminal Services

Once the installation is finished just follow these steps to install a terminal server:

1. Go to Control Panel | Add/Remove Programs | Windows Components and select 'Terminal Server'. Click 'Next'.

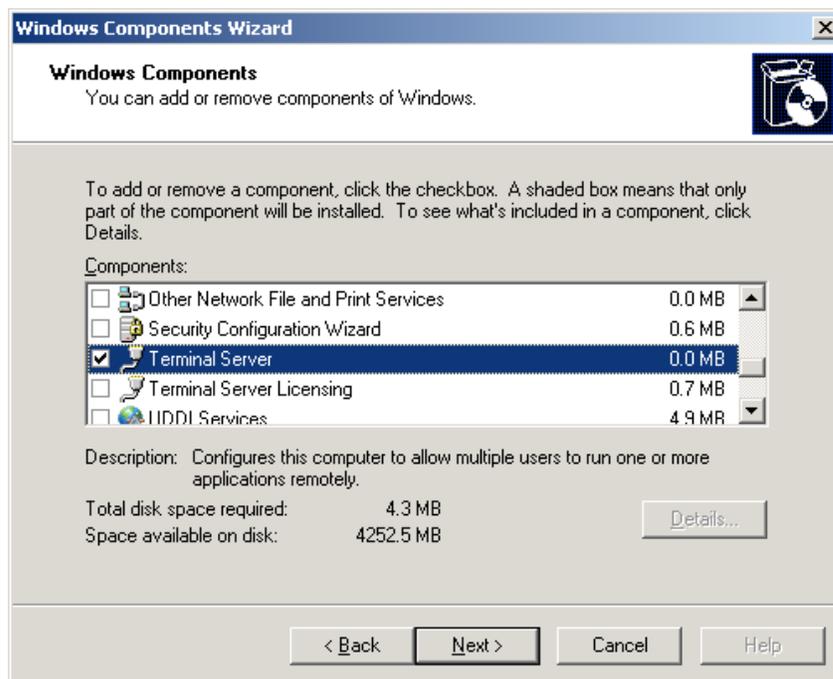


Fig. 24
Adding Terminal Server

2. Read the note about Terminal Server and click 'Next'.

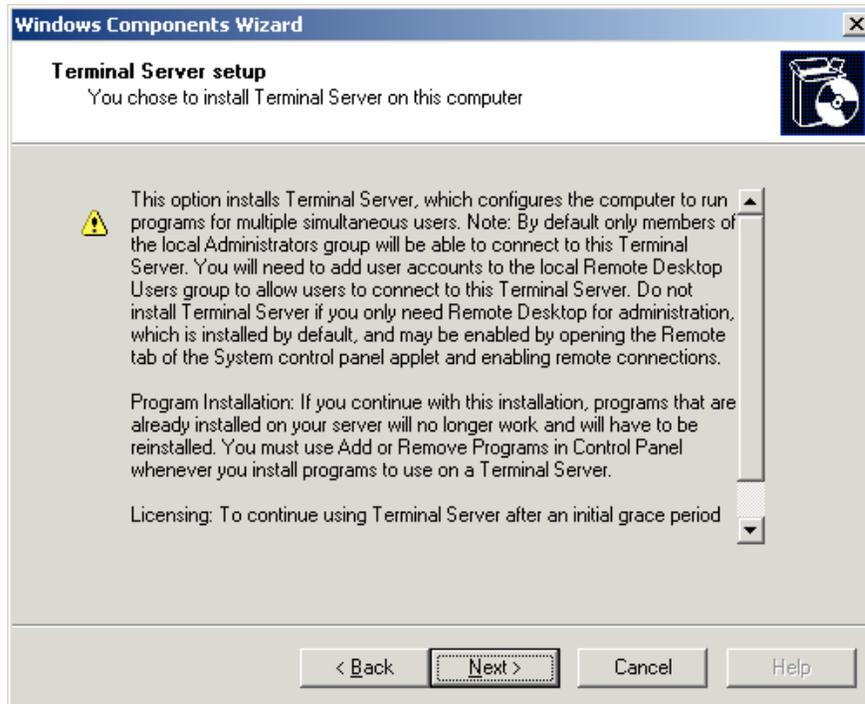


Fig. 25
Information about Terminal Server

3. Now you must select the security mode for your Terminal Server. The different here is simple to understand. When in 'Full Security' mode the TS will deny access to certain folders and registry keys that usually will not be used by a well written application or the most recent ones; the problem is certain applications may not follow best practices and/or may be old. In this case they may not work under TS if the security is set to 'Full Security'. As you can imagine the 'Relaxed Security' setting will allow access to these resources and the application will probably work. Based on that you may assume the best way is to set to 'Relaxed Security'. Well this is not the case. The best practice is to always set to 'Full Security' and if you find an application that does not work, set this to 'Relaxed Security' and try the application again (you can change this setting at anytime by launching TSCC.MSC on the TS and going to 'Server Settings' | 'Permission Compatibility'). If it works, the issue is permissions related. Set the server back to 'Full Security' and using tools like REGMON and FILEMON (freeware from Microsoft) find which resources are being denied and give users rights to these. This way you do not reduce security on all levels but only on the specific ones required by your application.

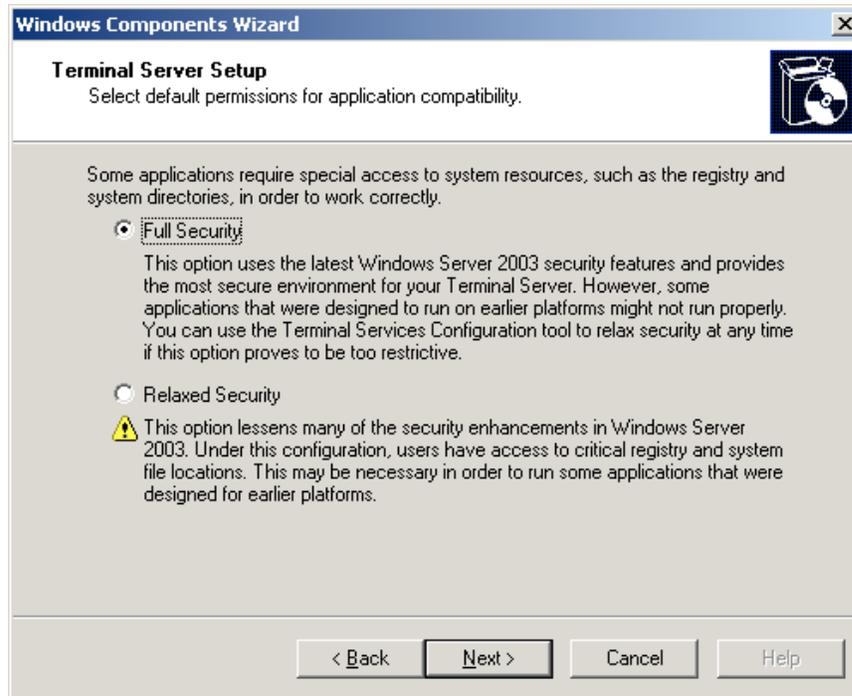


Fig. 26
Terminal Services Security

4. Now you need to specify the licensing server to be used. As we already installed our licensing server simply type the licensing server name (or IP address) and click the 'Check Names' button. If you see a message saying 'License Server Names are valid' you are ok to proceed. Otherwise check your licensing server and make sure it is installed, up and running. Then click 'Next'.

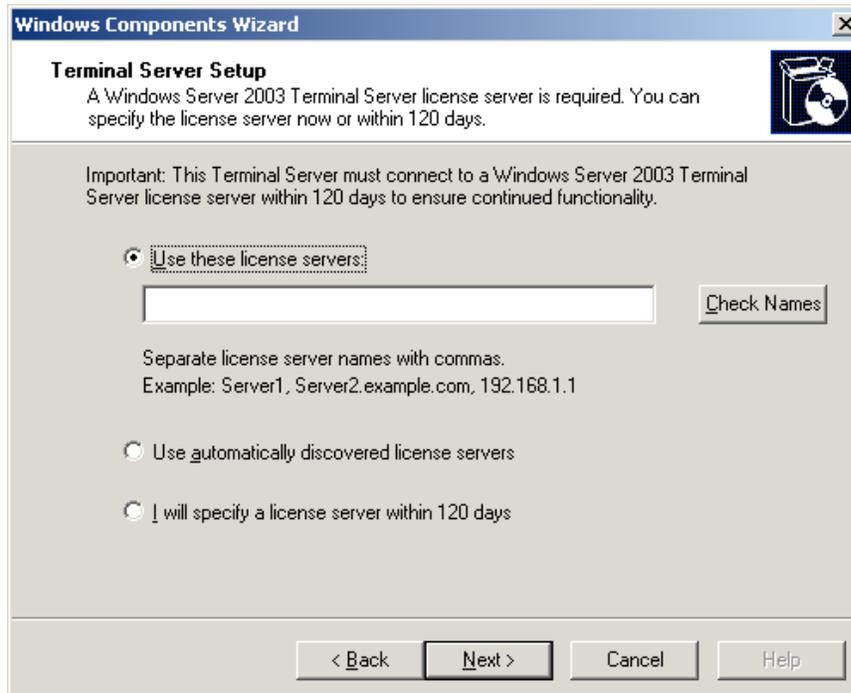


Fig. 27
Defining the licensing server

5. Based on what you learned previously regarding TS licensing modes choose the best option for your particular case and click 'Next'.

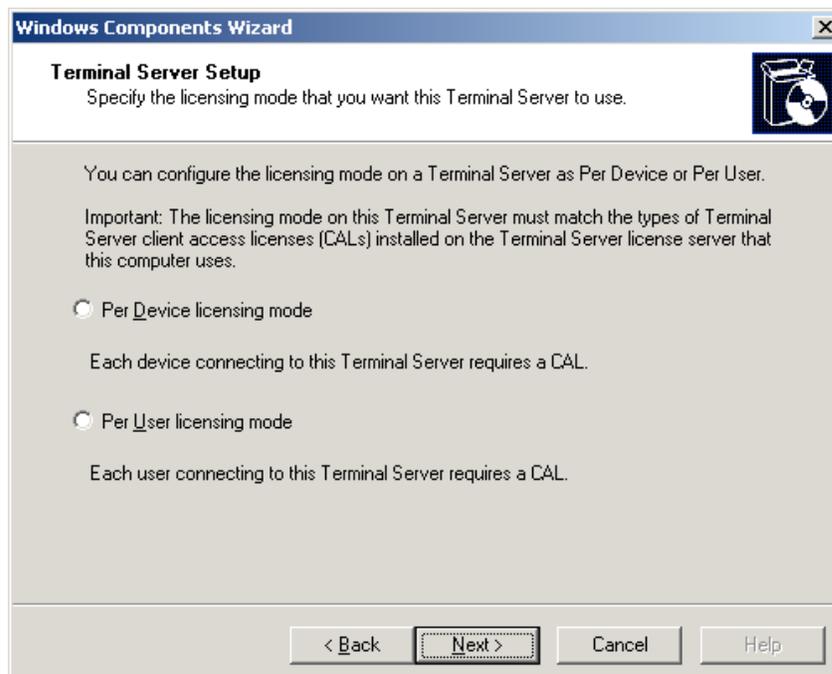


Fig. 28
Choosing the TS licensing mode

6. Just point to the media location where your Windows Server 2003 media is if asked and after retrieving all the required files you will be done! Just click 'Finish' to continue. The wizard will ask you to restart the server. Click 'Yes' and the server will restart. Congratulations, you just installed your first terminal server.

Allowing User Access to your Terminal Server

The next step is to install your applications. But before doing this you must add the 'TS Users' group we created a couple pages back to the 'Remote Desktop Users' local group on the TS. This will allow users on the 'TS Users' group to logon to your terminal server.

The required steps are:

1. Log on to your terminal server using an account with administrative rights.
2. Right click 'My Computer' and select 'Manage'.
3. Under 'System Tools' | 'Local Users and Groups' | 'Groups', double-click 'Remote Desktop Users'.

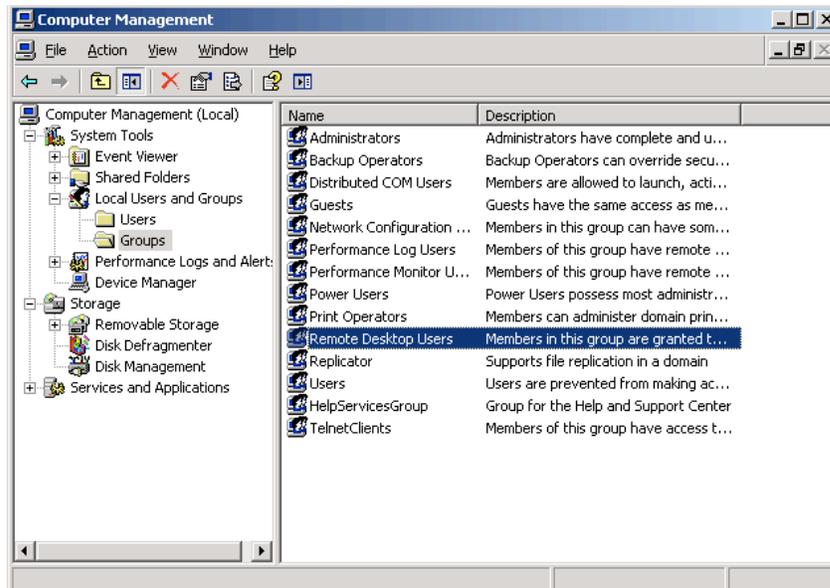


Fig. 29
'Remote Desktop Users' Group

4. Click 'Add' and add the 'TS Users' group that you created at the beginning of this guide.

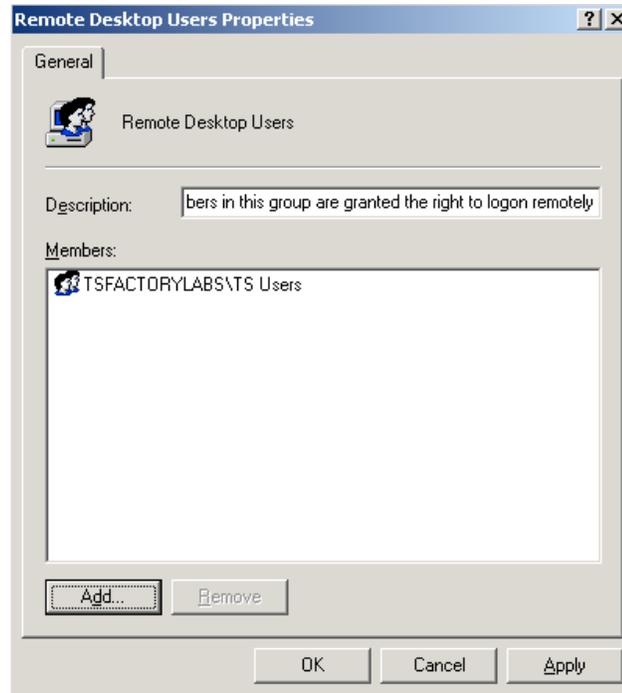


Fig. 30

Adding the 'TS Users' group to the 'Remote Desktop Users' local one

5. Just click 'Ok' and you are all set.

Installing Applications

If you have followed all the steps explained on the previous sections, by now you should have a working TS and users on the 'TS Users' group should be able to logon to it remotely. Well not really as I did not explain how to access the TS yet.

And assuming you have a purpose for your TS, there is a big chance you are setting all this up to provide users access to one or more applications either at the office or from home. If this is indeed the case we must go ahead and install applications. The main point here is very simple to understand: in a way TS is no different than any other regular PC and therefore installing applications is not really rocket science. But given the way TS works, there are some small differences that you must pay attention to. By following these simple rules (and understanding them) will save you time and headaches with TS in the future.

Basic Concepts

TS works as a multiuser 'workstation'; multiple users, at the same time, logon to the TS and run applications. This may introduce a problem with these applications as in many cases they were not developed with TS in mind (meaning multiple instances of the application, running under different user

accounts, at the SAME time on the SAME machine). To alleviate such problem, TS has something called 'Install Mode'.

This mode is triggered when you install applications using Control Panel | Add/Remove Programs or if you open a command prompt and type CHANGE USER /INSTALL before installing the application. When this mode is triggered, the TS will track all registry entries and files created by the application to determine what users need to run it, on a per-user basis (i.e. if an application uses a .INI file under \Windows, the TS will copy such file under the user profile so each user has its own, unique .INI file).

i When installing an application, if the application asks you for a reboot, do NOT reboot the server until you put the server back in EXECUTE mode. If using the command prompt you can trigger this mode by typing CHANGE USER /EXECUTE; when using Control Panel | Add/Remove Programs, simply choose not to reboot the server and click 'Next' on the wizard that will show up when installing applications. Just as a side note, all the information recorded when installing an application in install mode is stored under HKLM\Software\Microsoft\Windows NT\Current Version\Terminal Server\Install.

This is the main reason why it is very important to install applications using Control Panel | Add/Remove Programs is to allow the TS to track whatever is needed to make the application multiuser aware. If you follow this simple rule you will avoid a lot of issues with your TS setup.

Application Control

Now that you have applications installed on your terminal server, how do you control access to them? How to make sure users do not run applications they do not have rights to? That is exactly what Application Control means. Giving access to the right applications to the right users.

i One of the steps usually required to do this is to use Group Policies to redirect certain folders (i.e. Start Menu) on the system to a network location. As this we will be covering this later on this guide, all group policies related steps will have to wait for now.

I am assuming at this stage that your users will be connecting to the TS and they will be presented with a Windows Server 2003 desktop. Once that is in place users will then go to the Start Menu to launch the applications they have access to.

The first step I usually recommend is to create application groups on AD (i.e. MS Office Pro, MS Office Std, SAP7, MS Project, etc) and also language groups (if you will be using MUI to deliver different languages for the TS desktop, all

coming from the same TS; in this case create groups like TS English Users, TS French Users, etc).

Some people prefer to create department groups instead (i.e. TS Finance, TS Sales, TS Engineering, etc). Choose whatever you think works best for you. Here, there is no right or wrong way of doing things: both will achieve the same results but one may be easier or make more sense for your particular needs.

The idea here is very simple. Once you have these groups, the first step is to assign NTFS permissions to the application executables only to the groups that need access to that particular application (i.e. assign read/execute rights to the WINWORD.EXE file only to the MS Office Pro and MS Office Std groups; this way if someone is not on this group they will not be able to launch such executables). There are several ways to do this. There is always the manual way, that works well if you have one or two TSs. As these are really permissions being assigned to groups, you can easily script something that will use CACLS.EXE or XCACLS.VBS to set these. For more information please check this article:

How to use Xcacls.vbs to modify NTFS permissions
<http://support.microsoft.com/kb/825751>

The second step of course is to change the start menu based on the group membership so users will only see the applications they have access to.

If you follow these two simple steps you will end up with a well setup TS that will save you a lot of troubles down the road.

Application Troubleshooting

If you have made it this far, by now you should have at least one working TS, with applications locked down to run only by users that have access to them.

The next step is to understand what can go wrong with applications in a TS environment and where to look at when issues arise. This section will not make you an application expert but will give you a very good understanding on how to find issues on a TS environment and how to fix these.

You may think there is a lot to be covered here. Nope, that is not the case. TS is in a way, extremely simple and once you get the hang of troubleshooting applications you will notice how simple (not to mention repetitive) these steps are!

Tools

Application issues in TS are usually related to permissions. The typical example is an application that works when you logon as administrator but does NOT work when you logon as a regular user. Before we take a look at the tools you need to find where the problem is, let me refresh your memory with one simple thing. Do you remember when installing the TS, the step where it asked you if the TS was supposed to be in 'Full Security' or 'Relaxed Security'? If you do remember, this is the first, simple step you can take in this case.

If the TS is set to 'Full Security', launch TSCC.MSC on it and change it to 'Relaxed Security' and then ask the user to logon and try running the application again. If it works you know for sure it is permissions related. And that is when the right tools come to the picture.

This is the basic toolbox you should have to troubleshoot applications under TS and I will briefly explain each and how to use them.

REGMON: a simple and powerful tool that monitors everything that is going on in the registry (reads, writes, etc) and the results of such actions. For example if an application is trying to read a registry key and the user has no access to it you will see an 'ACCESS_DENIED' message on the regmon log. You can download it here:

<http://www.microsoft.com/technet/sysinternals/processesandthreads/regmon.msp>
[X](#)

FILEMON: the file system equivalent of Regmon. Every time something accesses the file system Filemon will show that and the result. If an application is trying to read a file the user has no access to, an 'ACCESS_DENIED' message will be shown in the logs. You can download it here:

<http://technet.microsoft.com/en-us/sysinternals/bb896642.aspx>

 Regmon and Filemon were combined into a single tool, now called Process Monitor. It replaces both tools but the idea remains exactly the same. You can get it here:

<http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>

Now that we have the tool (let's use Process Monitor) let's come up with an example to show how to use them. On this test I will change the NTFS permissions for NOTEPAD.EXE and will deny access to my test user. By using Process Monitor we will be able to see the issue and how to fix it. The steps we will follow should be applied when troubleshooting any application under TS.

1. From your client PC launch MSTSC and connect to the TS using an account with administrator rights.
2. Launch Process Monitor. Press CTRL+E to stop capturing events and CTRL+X to clear the log.

3. Logon to the TS as the user account having issues (in our example, the user that cannot use Notepad for some reason).
4. Before launching Notepad as the user, go back to your Administrator session and start capturing data with Process Monitor (use CTRL+E).
5. Go back to the user session and try to launch Notepad. You will see an error message.



Fig. 31
Error launching Notepad as a regular user

6. Go back to your administrator's session and press CTRL+E to stop capturing additional data. Now press CTRL+F to bring the 'Find' window up. Type 'denied' and press Enter.

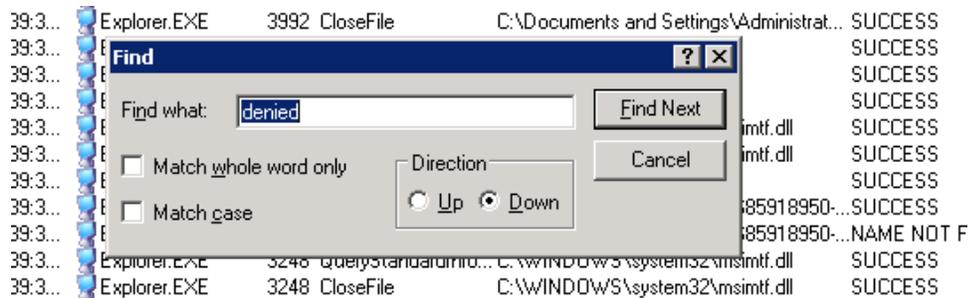


Fig. 32
Looking for an 'Access Denied' error

7. As we can see the problem is the user has no rights to Notepad.exe. To fix the problem we just need to check the NTFS permissions on the file and make sure the user has rights to it!

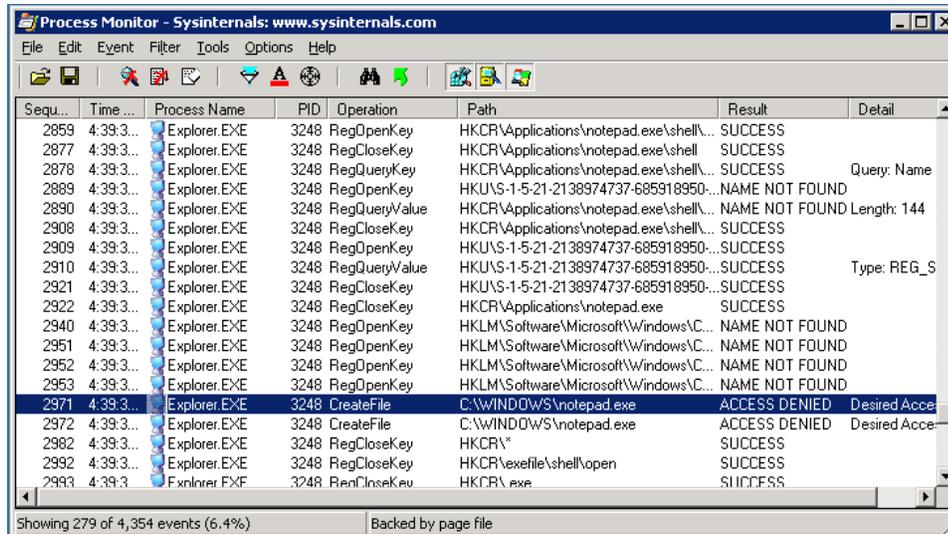


Fig. 33
'ACCESS DENIED' error on Process Monitor

The idea in this section was not to make you an application expert overnight and I am sure that is not the case right now after reading this far; but with the steps and tools described above you will be able to troubleshoot and fix almost all application issues that happen in a terminal services environment.

i One thing that is definitely worth mentioning is the use of 'Application Flags' when trying to fix certain issues. One typical example is an application that works when the TS is in install mode (remember, CHANGE USER /INSTALL) but not when in execution mode (after you issue the CHANGE USER /EXECUTE command, once the application install is complete). This usually has to do with the application looking for certain files under C:\Windows (the real Windows directory) and not the redirected one that TS uses for users (under their profile/home directory usually). Once you set a flag for this (0x00000080) everything starts to work properly!

For more information on all the available flags and how to use them please check this article:

<http://technet2.microsoft.com/windowsserver/en/library/df78c476-00d5-41f0-a21d-e1e12e3d1f8b1033.mspx?mfr=true>

Accessing the TS

By now, as you had to connect to your terminal server to make sure it was running and to get familiar with Process Monitor as per above, I am sure you know how to connect to the TS.

But as there are a couple of ways to do that, I think it is a good idea to explain what each client is, what they do and how to use them.

Full client

By full client I mean the full, Win32 client that comes with Windows XP or Windows Vista (the famous MSTSC.exe one). Of course there are different versions of such client (for RDP5.0, RDP5.2, RDP6.0 and now RDP6.1).

At this time the latest RDP client can be downloaded at:

<http://support.microsoft.com/KB/925876>

As you already know, to access a terminal server using this client is a very simple process. Just launch MSTSC (go to Start | Run | MSTSC and press enter) and enter the IP address or FQDN of your TS. If you click on 'Options' all the available options (drive mapping, printers, etc) will appear.

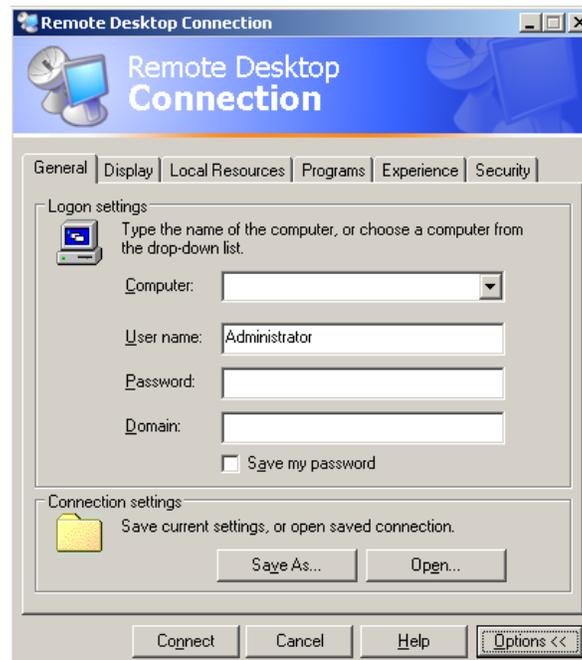


Fig. 34
Remote Desktop Client

There are also some command-line options (like /console to connect to the 'console' on your 2003 box). Just type `mstsc /?` to see all the available options.

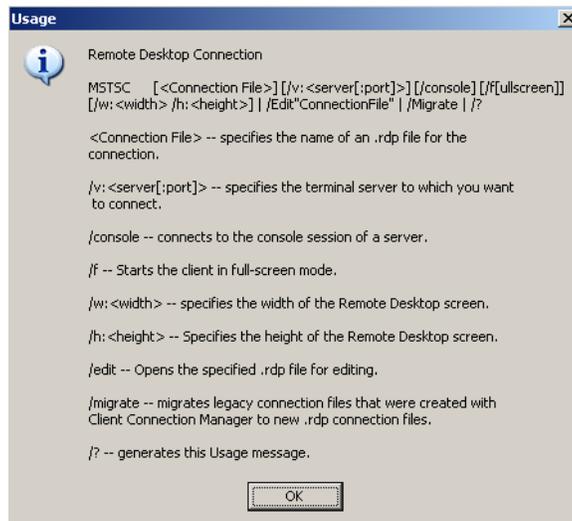


Fig. 35
MSTSC command line options

This is the client in use on Windows XP Embedded thin clients and on Windows Server 2003 as well. Of course they may have different build numbers (i.e. 5.2.3790.1830, etc) but we usually refer to them as the 'Win32' client.

Web client

A couple months after releasing Windows 2000 back in February, 2000, Microsoft released an ActiveX version of the RDP client. For the first time users were able to connect to a simple web page and from there, with the ActiveX client automatically loaded on their machines, connect to a terminal server anywhere on their networks or on the Internet (to see how popular this became simply do a search on Google for "allinurl:tsweb/default.htm"; the results are impressive!).

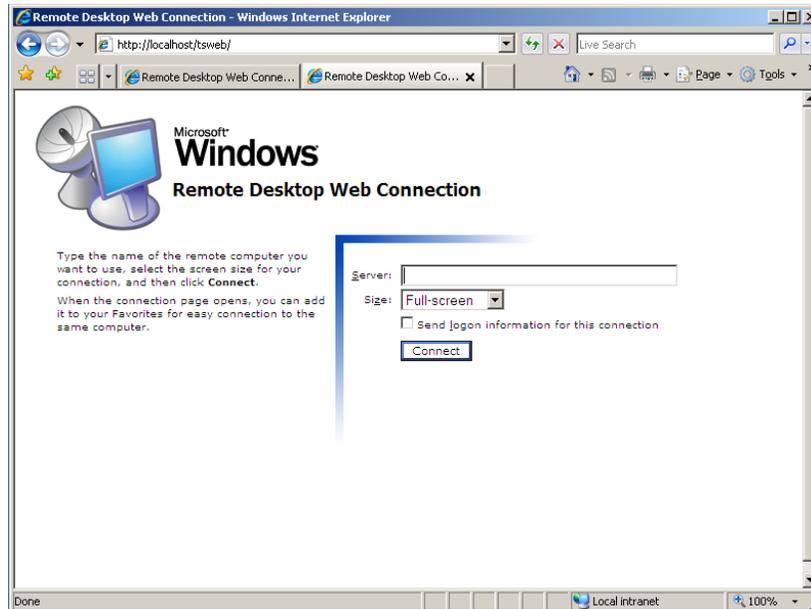


Fig. 36
TSWEB default web page

Although this is indeed a neat way to access your terminal servers, this also mislead people to thinking they could connect to their terminal servers through a single port (http:80 or https:443); in reality, tswb was simply a mechanism to deliver the RDP client through a web browser and to provide a front end for the client options (i.e. username, server name, resolution, etc) but the actual RDP connection was still going through port TCP 3389 so at the end two ports were required and as of today, this is still the case with Windows Server 2003.

Windows Server 2008 does have a mechanism to use a single port for RDP access, over https.

i Again, remember that tswb will not give you RDP over a single port like 80 or 443; even though your users may be able to access this port through their web browser, they still need to be able to reach the terminal servers through port TCP 3389. And of course I do not need to mention this page is NOT compatible with Linux and/or Mac OS X as it requires an ActiveX control (Windows only) to be loaded...

Setting up TSWEB is quite simple actually; all you need is a working IIS server.

Download the tswb package directly from Microsoft at:

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

To setup tswb:

1. Download the package listed above and copy it to your IIS server.
2. Double click tswbsetup.exe to start the installation.

3. Click 'Yes' to install the tsweb package.

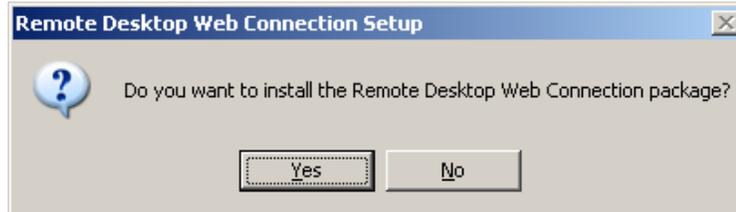


Fig. 37
Installing TSWEB

4. Click 'Yes' to accept the license agreement.

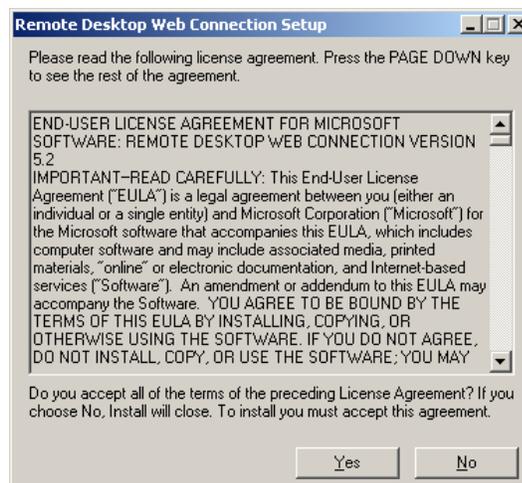


Fig. 38
TSWEB license agreement

5. Select the folder on your IIS server where you want TSWEB to be installed. I would recommend another folder as you could see on Google the amount of companies using the default page and exposing their infrastructure details on the internet!



Fig. 39
TSWEB default installation folder

6. Once the installation finishes it will ask if you want to read the release notes.

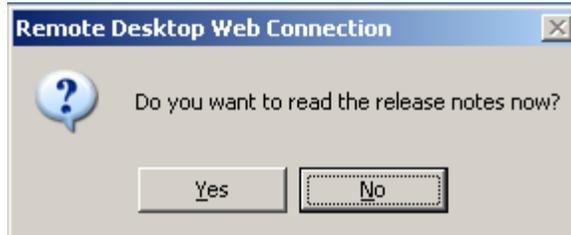


Fig. 40
TSWEB release notes

7. If you followed all the steps above you should see this window.



Fig. 41
Successful installation

Now all you need to do is to point your users to your web server and they will see the default TSWEB webpage. If they are connecting over the internet, remember you will need to make sure your TS can be reached (port TCP 3389 open) and that they use the TS FQDN or external IP address to connect!

Other clients

Of course the next question is 'what if my clients do not run Windows? Can they still connect to my terminal servers?'. The answer for this question is 'Yes, they can'.

The only non-Windows platform supported directly by Microsoft is Mac OS X. A native OS X client (universal binary!) does exist and is available for download on the Microsoft website.

You can get it here:

<http://www.microsoft.com/mac/downloads.mspx>

For Linux the most common client is RDesktop, a free, open source alternative. As RDP was a proprietary protocol until March, 2008, all non-official RDP clients lacked support for some sort of feature (i.e. proper serial port redirection) and RDesktop was one of these. The same is valid for some Java implementations

out there. Therefore if your most important requirement is a fully compatible RDP client, as of today your only alternative is to use the full Win32 one.

But this did not prevent other companies to develop RDP clients and replacements for the desktop OS with a built-in RDP client, basically turning PCs into Thin Clients with a very light OS that can be loaded on the machine over the network (using PXE boot) or even from USB drives or CDs. One of the most impressive solutions out there is the 2X ThinClientServer. It not only allows you to boot PCs with its own streamlined OS but can handle real thin clients as well. All this from a nice web based console.

i Note: I am not mentioning the 2X solution because they sponsored this guide. I have actually tried many similar solutions and years ago had my own distribution to do the same. But none of them, including mine, had all the centralized management features on top of a pretty web console. Add to that the fact they have a free version (the unsupported PXES one) and I still think there is no other solution like this on the market as of today. I know this may change in the future, like with any other software solution out there. But as of today they are 'The Solution'.

You can check them out at <http://www.2x.com>.

It is worth mentioning that when using thin clients (small computer like devices with little local processing power and designed to be used as a 'dumb' terminal, connected all the time to a terminal server) you must pay extra attention on your requirements. As the OS on these devices vary (i.e. Windows XP Embedded, Windows CE, Linux, etc) the RDP client on these will not be the same across the board; this means certain features may not be fully supported on the device depending on the OS it is running. If your main concern/requirement is 100% compatibility with the latest RDP version out there your best bet is to use devices running Windows XP Embedded.

Printing

If you take a look at the Microsoft public newsgroups or at the Remote Desktop/Terminal Services forum at Experts-Exchange you will notice that many questions (if not most of them) are related to printing. The same is valid for TS add-ons like Citrix; printing issues do exist and have plagued TS for many years.

The main question is, is it really that bad? Well it all depends on what you do on the TS, if you follow best practices, if you control the other end (meaning the user end and the printers they may have) and so on. We will give you a very good understanding on what you can do to minimize these and in some cases, eliminate such issues completely.

Best practices

The first thing you must learn before we move ahead is quite simple and easy to remember: **NEVER**, and I mean **NEVER**, install any printer driver on your terminal server. Ok, you can install them but only if the driver is developed for Windows Server 2003, it is certified by Microsoft **AND** (this is important!) you tested in your test environment with at least 5 to 10 users connected to the TS and all trying to print different types of documents, ranging from slides to 100 page PDFs. If the driver meets all above it is probably safe to have it running on your TS.

The second thing to remember is to avoid host based printers (these are printers that rely on the driver running on the machine to perform several tasks that are usually handled at the hardware level, directly on the printer – manufacturers do that to reduce the printer costs) and multifunctional devices (those that print, scan, fax, make coffee, etc – all in one single unit). I know this is probably the hardest part, as usually we have no control of which printers our users will be using at home for example.

The final rule is the easiest: avoid using/buying/suggesting less known brands and models.

If you follow these three simple rules above you will cut down a lot of your printing issues. If these still happen, let's see what can be done.

Alternatives

There are two different types of alternatives out there. Free workarounds and paid solutions. Note the difference between workarounds and solutions. Workarounds will probably get you printing but functionality may be lost along the way (i.e. being limited to black and white printing even though your printer is a fancy color one); solutions in the other hand will give you all the features you need and more (i.e. bandwidth control/compression) but with a major drawback: cost.

Workarounds

Before we go ahead and discuss the known workarounds out there, you must understand how printer mapping works under Terminal Services. It is actually quite simple. Once a client connects to the TS, the TS will retrieve the exact name of the printer driver on the client end (under the printer 'Advanced' tab) and will compare it to what is available on the server. If a match is found, the TS will use the driver it has installed locally (if you did not install any drivers this means the Windows Server 2003 out-of-the-box printer drivers) and it will create a printer under the user session (you can easily identify these as they have the word 'session' as part of its name).

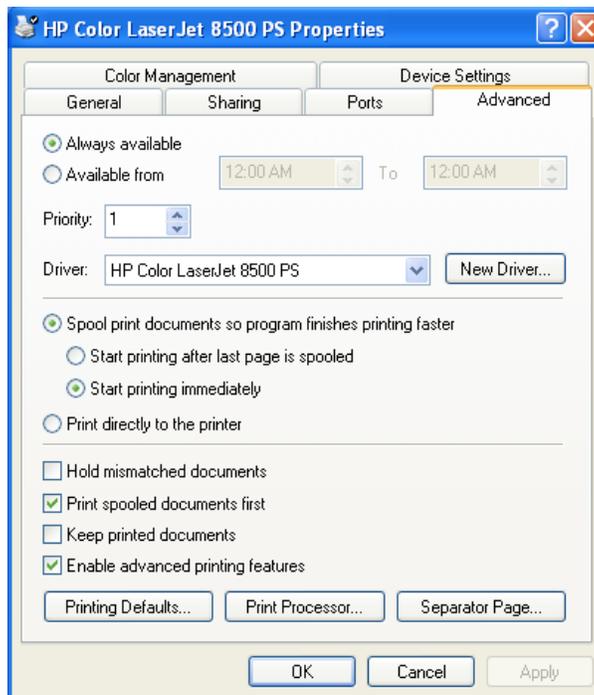


Fig. 42
Printer properties on the client side

As you can see, if you use printers that are available out-of-the-box with Windows Server 2003 (to find out which ones are there, simply logon to the server and add a new printer – use the LPT1: port or any other port as this is just for determining models available – and then browse through the manufacturer's list and models), your local printer will get 'mapped' under your terminal server session and you will be able to print right away. This happens by default with any Windows Server 2003 default installation.

What if I am using a printer that is not on the list? As we mentioned above we should avoid installing any drivers on the TS. So how can the TS map my printer if the drivers names do not match? Well someone thought about that and added a mechanism to handle these cases under Terminal Services...

And the solution, as always, is simple. Here are the step-by-step instructions:

1. You bought a LexBrother printer model CHEAPO1000. You installed it on your local PC and it works. The driver name listed on the 'Advanced' tab is 'LexBrother Cheapo 1000 Series'.
2. The first thing to determine is if this printer is compatible with anything else. Usually laser printers are compatible with some HP LaserJet model (like the old LaserJet 5 ones) and inkjet ones may be compatible with HP DeskJets (if you see another model under the Windows Server 2003

default printer list from the same manufacturer as your printer, you can try that driver to see if it works). Assuming there was no Windows Server 2003 for our CHEAPO1000 and that it worked with a LaserJet 4 driver we are ready to go. This step is where you will spend most of your time: trying to determine which driver may work with your printer.

3. On the TS, edit the file PRINTUPG.INF and add a line under the [Printer Driver Mapping] section as shown below (yes, it is case sensitive):
"LexBrother Cheapo 1000 Series" = "HP LaserJet 4", 1, 1, "11/27/1999"
The left column is the exact name of the local driver, as seen on "your PC and the right column is the exact name as shown under the printer list available on Windows Server 2003 plus some extra values (check the PRINTUPG.INF file on your 2003 Server for more information – usually all you do is to copy an existing line and paste it right below and then you just change the left column to match the client driver name).
4. Now you must change the TS registry. Add two new values to this registry key:
HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\rdpwd
The values are:

Name: "PrinterMappingINFName"
Type: REG_SZ
Data: C:\Windows\Inf\PRINTUPG.INF.

Name: "PrinterMappingINFSection"
Type: REG_SZ
Data: Printers
5. From now on the TS will use the LaserJet 4 driver anytime someone connects using the CHEAPO1000 printer!

i Note: If you do want to change the PRINTUPG.INF file you can create your own .INF one. Just follow this article (it also details all I explained above):

<http://support.microsoft.com/kb/275495>

"Printer redirection or upgrade may not work because of signed Ntprint.inf file"

Some other things to remember when troubleshooting printing issues:

- Make sure you have the latest RDP client on your machine.
- If you use a Mac try using a postscript printer.
- For older clients check Q302361 in the Microsoft Knowledge Base.

Third Party Printing Solutions

The purpose of this section is not really to evaluate and write a review of every single product to do this out there. I just want you the reader to be aware that such products exist, explain what they do and give you an idea of how much they cost and why they are helpful.

I have seen environments where administrators and/or technical personnel would spend hours and hours per week troubleshooting/fixing printing issues in TS. If you consider all these hours had an actual cost to the company that employed all these people we can easily see these companies were spending thousands of dollars monthly on printing problems. The issue is, most administrators/techies do not see their time as an expense, which is completely wrong.

And that is why I think all these products are helpful. They eliminate or greatly reduce printing issues and give administrators and techies time to do other, more relevant things for their companies. The problem is, there is an upfront AND steep cost usually associated with such solutions. I do not have exact numbers but they range from US\$1,000 to US\$ 2,000 per terminal server (unlimited users). If you are planning a TS environment, I highly recommend you budget some extra money for add-ons you will probably need to get the job done properly. Make sure printing is covered in this budget.

Not getting into the technical details of each product, the main idea behind all of them is eliminate the need for any drivers on the TS. The job is sent to the client and the client, using any printer it may have, will print it. There is also a way to control how much bandwidth is being used for printing (to limit it so you do not take over the whole DSL link just for your print job!) and compression. Neat stuff for sure. Keep in mind a client component is required (meaning you will have to deploy something at the client end for these products to work).

The products I know, in no specific order, are:

- UniPrint
- ThinPrint
- SimplifyPrinting
- Print-IT

If printing becomes an issue or if you do not have control over what users may use as printers I definitely recommend you taking a look at any of the products above.

Terminal Services Security

Now that we have a terminal server up and running and with applications installed on it, the next step is to learn how to make your terminal server more secure.

The term 'secure' in this context has several meanings; for example, it could mean securing the TS from hackers or locking it down so your own users do not access things they should not have access to.

Group Policies

I think the most important step to learn with terminal services is how to properly use Group Policies. Usually what we want to achieve is to apply certain restrictions to our users but only when they are connected to the TS servers and NOT when they are at their workstations. For example you do not want your users to see the terminal servers system drives (i.e. C:) or to be able to shutdown the server but if they are logged on to their PCs they should be able to see these drives and turn off their computers!

The way to do this is actually quite simple: all you need to have is a group policy but with a little setting enabled: the loopback processing mode. This will achieve exactly what we want. Users will get restricted when on the TS but not on their PCs. So let's take a look at how to implement such policy. I assume you have the TS up and running and part of a domain – preferably Windows Server 2003 AD – and that you have rights to do what we will be discussing next.

Step-by-Step procedure:

1. Logon to your domain controller and launch 'Active Directory Users and Computers'. If you followed the steps described under 'Active Directory Preparation' at the beginning of this guide you should have an OU called 'Terminal Servers' with your TS computer objects.

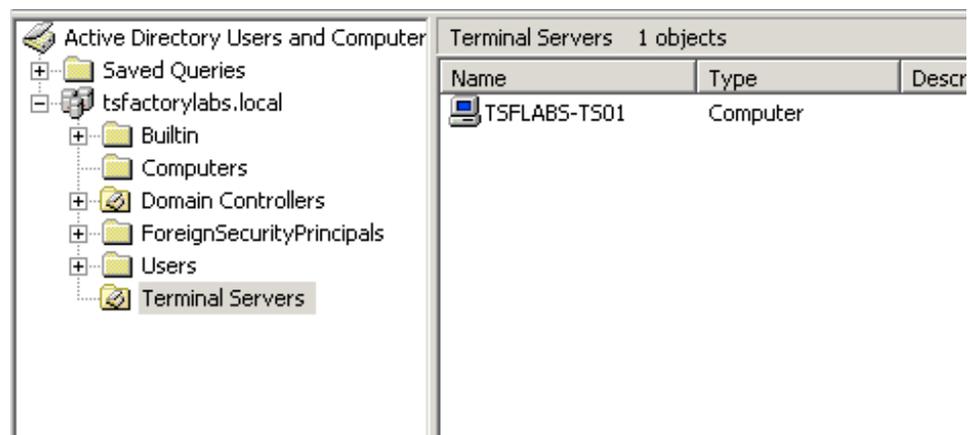


Fig. 43
Active Directory Users and Computers, Terminal Servers OU

2. You should also have the two groups we created way back at the beginning of this guide: 'TS Users' and 'TS Servers'. Make sure these exist and that you have your users and servers into the respective groups.

3. Right-click the 'Terminal Servers' OU and click 'Properties'.
4. Click on the 'Group Policy' tab and then click 'New'. If you are using the Group Policy Management Console you will need to launch that tool and navigate to the 'Terminal Servers' OU and create a new policy there.

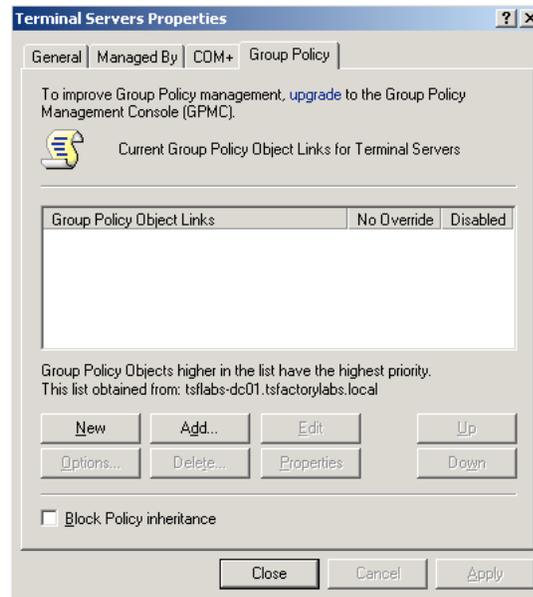


Fig. 44
Creating a new group policy

5. Give your policy a name which is easy to remember like 'TS Policy' and then click 'Properties'. Click on the 'Security' tab and there remove 'Authenticated Users'. Also make sure that for 'Domain Admins' and 'Enterprise Admins' you click the 'Deny' checkbox for 'Apply Group Policy'. Now add the groups 'TS Users' and 'TS Servers' and make sure the 'Allow' checkbox is checked for 'Apply Group Policy' for these two groups. Do not forget to click 'Apply'!

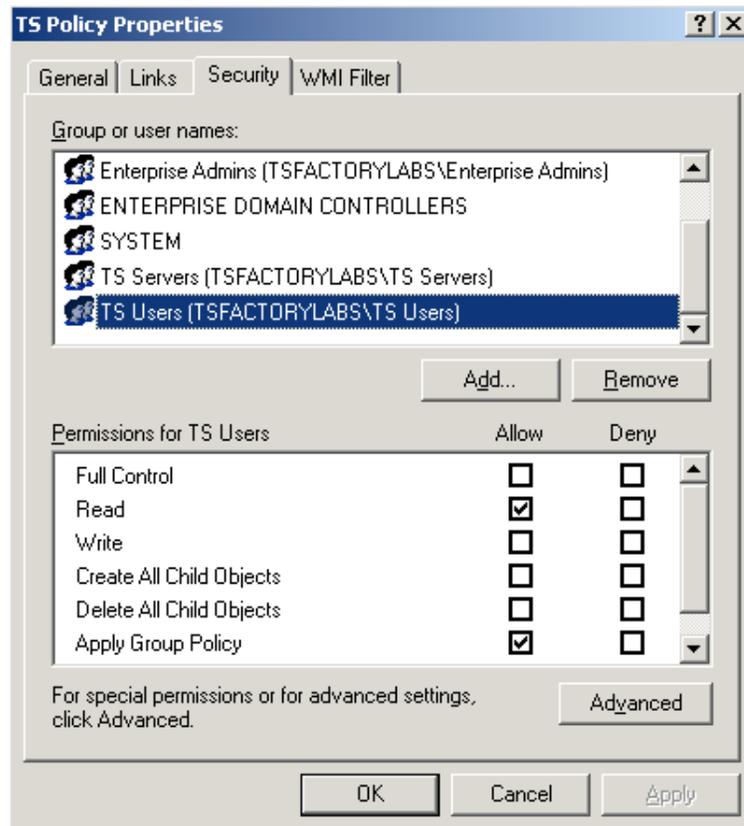


Fig. 45

'TS Policy' policy. Note the groups and checkboxes settings

6. Click 'Ok' to go back to the previous screen (where you see the policy name, 'TS Policy').
7. Now we need to test if the policy is working. Click on 'TS Policy' and then click 'Edit'.
8. The 'Group Policy Object Editor' window will appear. Under 'User Configuration' | 'Administrative Templates' | 'Start Menu and Taskbar' find 'Remove Run menu from Start Menu' and double click it.

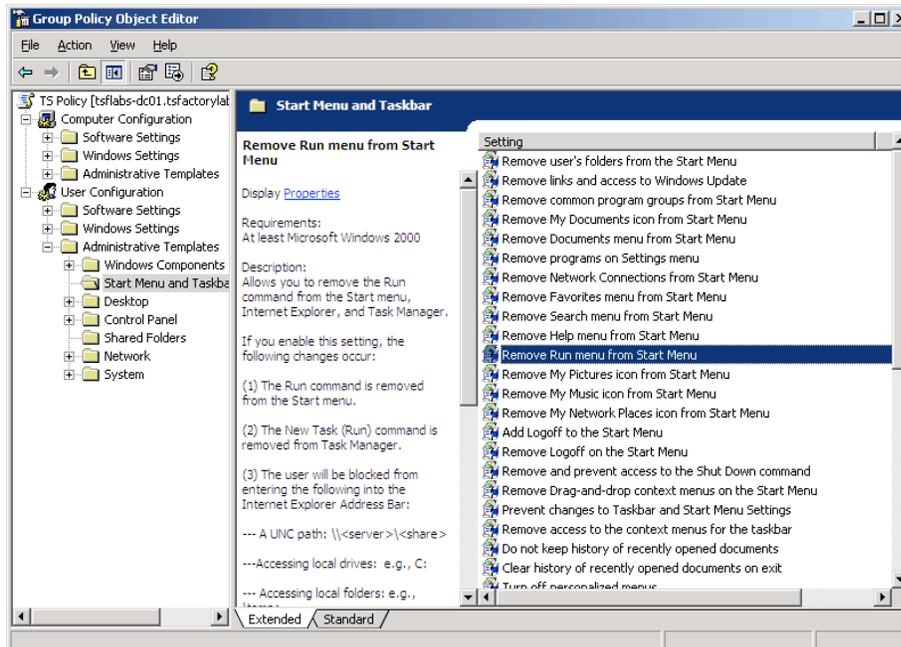


Fig. 46
Group Policy Object Editor – Remove Run from Start Menu

9. Set it to 'Enabled' and click 'Ok'.



Fig. 47
Enabling 'Remove Run from Start Menu'

10. Now the most important step: under 'Computer Configuration' | 'Administrative Templates' | 'System' | 'Group Policy', make sure you enable the 'User Group Policy loopback processing mode' and set it to replace.

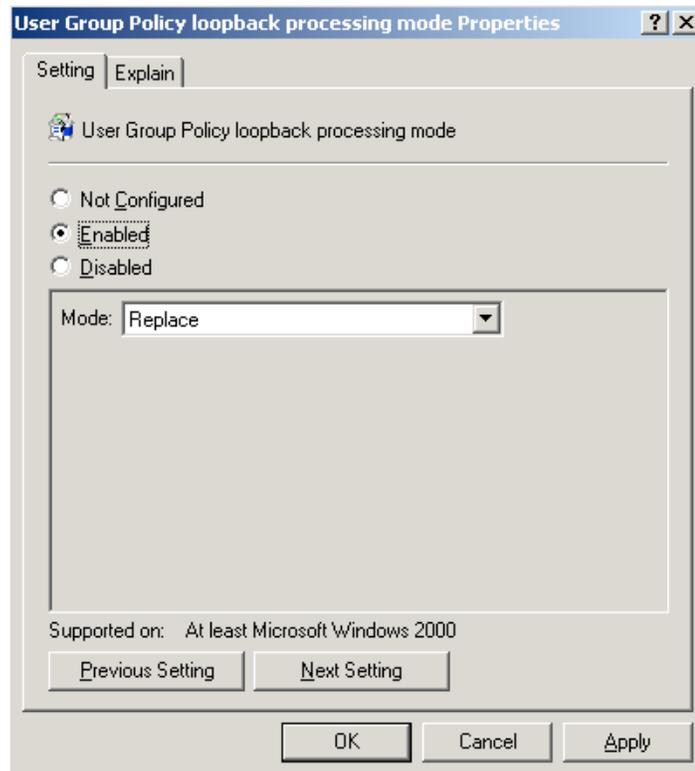


Fig. 48

Enabling the loopback processing mode

11. Close the 'Group Policy Object Editor' and then click the 'Close' button on the 'Terminal Servers Properties' window.

If you did everything right, we should be good to go. Reboot your TS and try to logon to it as a 'TS Users' group user. You should see a screen similar to this one, with no 'Run' option on the Start Menu!

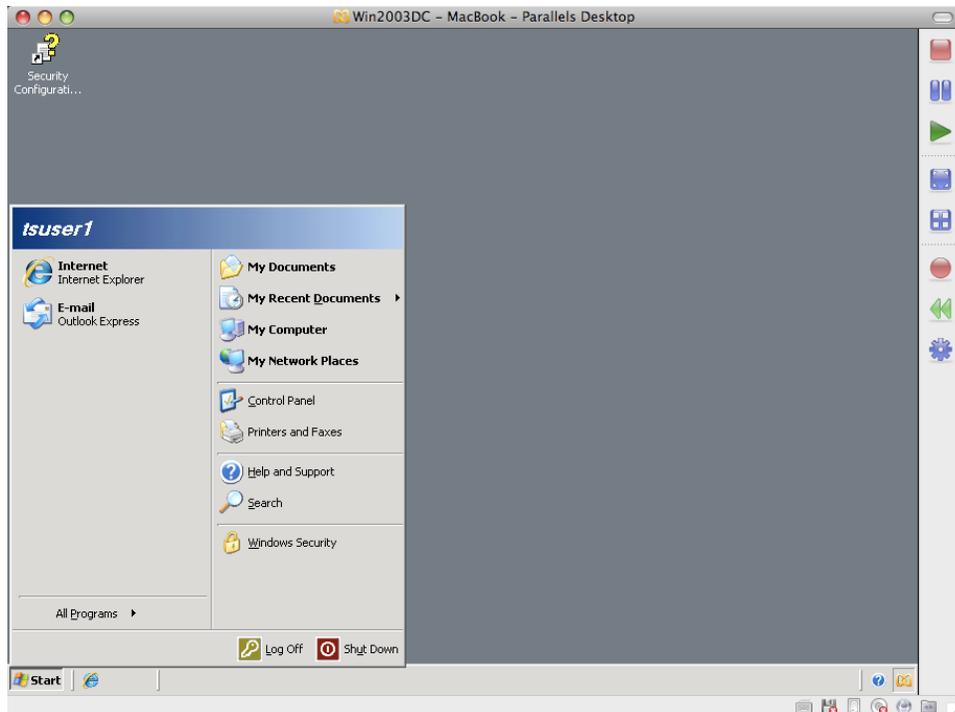


Fig. 49

No 'Run' for this user

Now that our basic policy is working we can start our lockdown process. Again, this is one of the areas where people will have different settings based on their needs. I will set some of the basic ones I think are relevant and you just go from there!

Lockdown

Most of the lockdown settings in a TS environment will be performed on the Start Menu/Taskbar and as well on Windows Explorer. As already mentioned, the settings I use on this guide are simply a start point and you should add more settings if needed in your particular environment. I would just like to mention that you should always add more settings one-by-one so in case something goes wrong or shows some unexpected side-effects you know exactly how to fix the issue!

We will basically continue from where we left our group policy settings. Just launch 'Active Directory Users and Computers' or the 'Group Policy Management Console' and open the 'TS Policy'.

The follow these steps to set some of the basic recommended settings:

1. Navigate to 'User Configuration' | 'Administrative Templates' | 'Start Menu and Taskbar' and set the following options:

- a. Remove links and access to Windows Update: Enabled
 - b. Remove Favorites menu from Start Menu: Enabled
 - c. Remove Search menu from Start Menu: Enabled
 - d. Remove Help menu from Start Menu: Enabled
 - e. Add Logoff to the Start Menu: Enabled
 - f. Remove and prevent access to the Shut Down command: Enabled
 - g. Remove access to the context menus for the taskbar: Enabled
 - h. Force classic Start Menu: Enabled
2. Navigate to 'User Configuration' | 'Administrative Templates' | 'Windows Components' | 'Windows Explorer' and set the following options:
 - a. Turn on Classic Shell: Enabled
 - b. Hides the Manage item on the Windows Explorer context menu: Enabled
 - c. Hide these specified drives in My Computer: Enabled; Restrict A, B, C and D drives only.
3. Navigate to 'User Configuration' | 'Administrative Templates' | 'Control Panel' and set the following options:
 - a. Prohibit access to the Control Panel: Enabled
4. Navigate to 'User Configuration' | 'Administrative Templates' | 'System' and set the following options:
 - a. Prevent access to the command prompt: Enabled; Disable the command prompt script processing also? No
 - b. Prevent access to registry editing tools: Yes; Disable regedit from running silently? No
5. Navigate to 'User Configuration' | 'Administrative Templates' | 'System' | 'Control+Alt+Del Options' and set the following options:
 - a. Remove Task Manager: Enabled
 - b. Remove Lock Computer: Enabled

These are the basic recommended settings. I will not explain every single option here for one simple reason: when you are selecting these you will be able to read the explanation for every option available. And, one more time, these are the basic settings I recommend and depending on what you need you may have to set extra options!

Remember that Office applications have their own ADM templates that you can import directly in the 'Group Policy Object Editor' window; simply right click 'Administrative Templates' (under 'Computer Configuration' or 'User click

Configuration' and select 'Add/Remove Templates'. On the next screen simply click 'Add...' and browse for the .ADM files.

Note: for Microsoft Office, simply download the Microsoft Office Resource Kit for the version of Office you are using and install it on your terminal server. The template files (.ADM) will be installed as part of the resource kit.

Assuming you set all the options above, this is how your user desktop will look like when he connects to the TS.

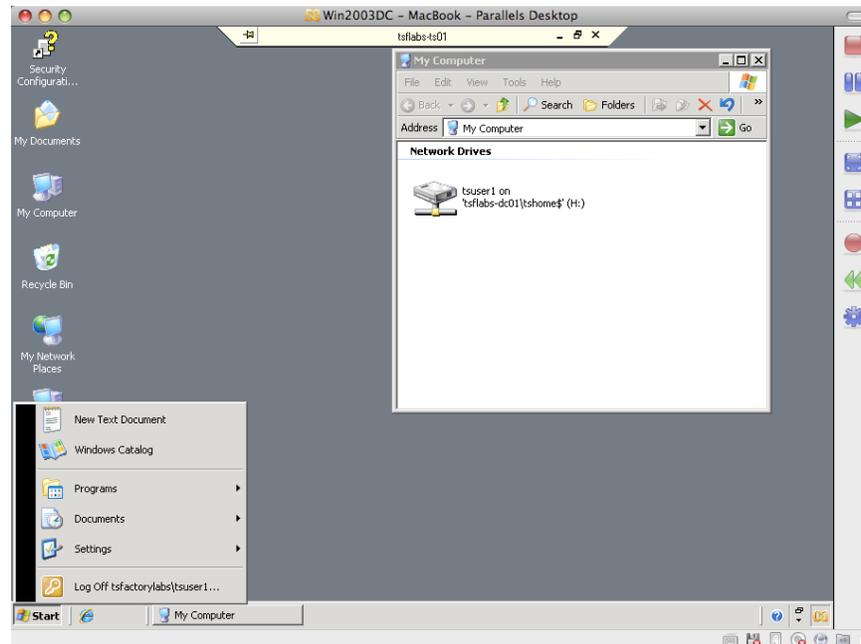


Fig. 50
A locked down desktop

Note that all server drives (C:, D:, etc) are hidden and the Start Menu had many options removed from it!

Folder Redirection

As part of the lockdown process, one important and very useful step is 'Folder Redirection'. The concept here is simple: once enabled, certain folders (i.e. Start Menu, Desktop, etc) are redirected to a network location for your users (by username, groups, etc). This is important for a couple reasons. For example users tend to save files to their desktop. When using a TS this may become a huge problem as by default such folder is part of the user profile and if you are using roaming profiles this means lots of data will be loaded every time the user logs on or logs off the TS which greatly increases logon times.

To preserve the user experience, look and feel, you probably want to give your users the same start menu, only showing the applications they have access to, regardless of the TS they are logged on. Start Menu redirection fixes this particular example.

So let's learn the basics on how to do this by following these step-by-step instructions. In this example we will be redirecting the Start Menu and the Desktop for our 'TS Users' group.

1. The first step is to create two folders on your file server and share them. I recommend you using a meaningful name. In my case I created 'StartMenu' and 'Desktops' and shared them as 'StartMenu\$' and 'Desktops\$'. The permissions must be set as shown below.
 - a. Desktops\$ share: Domain Admins, Full Control; TS Users, Full Control.
 - b. StartMenu\$ share: Domain Admins, Full Control; TS Users, Read.

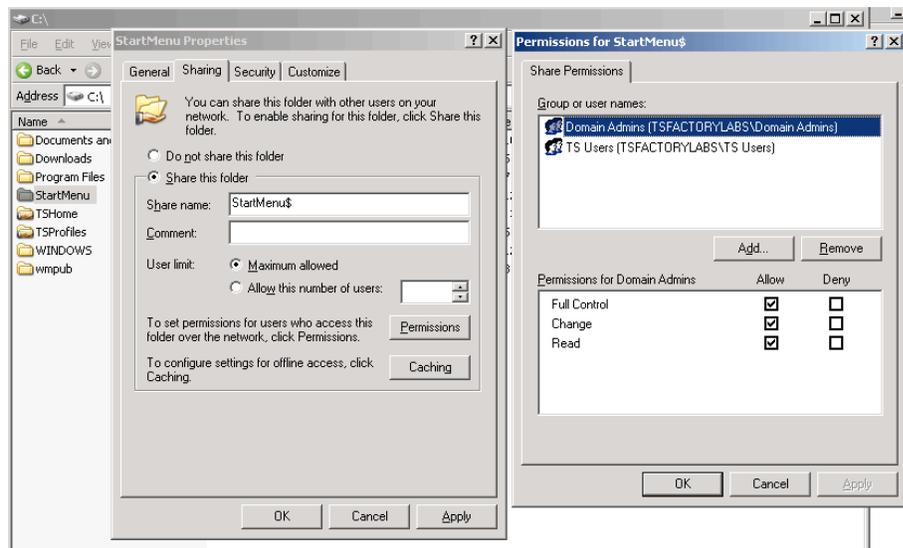


Fig. 51
StartMenu\$ share

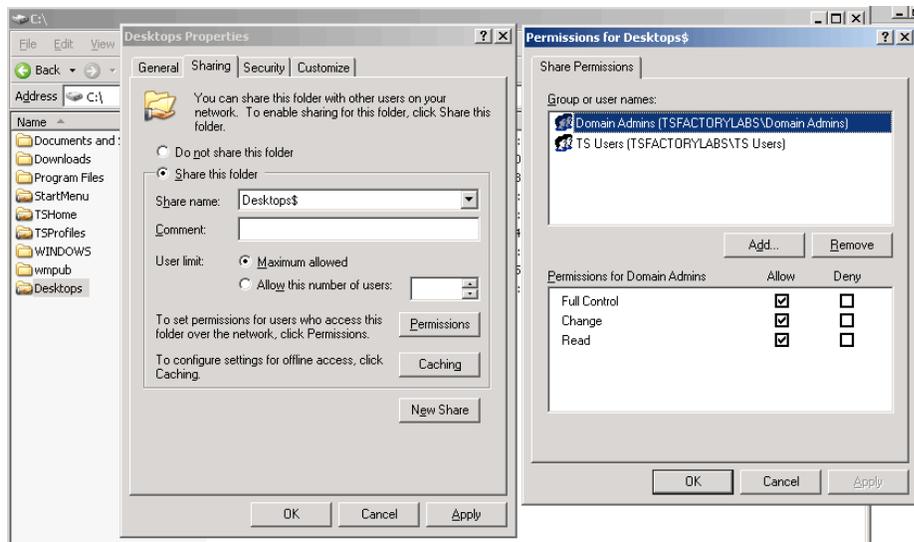


Fig. 52
Desktop\$ share

2. Now launch 'Active Directory Users and Computers' and go to the 'Terminal Servers' OU to open our 'TS Users' Group Policy. Navigate to 'User Configuration' | 'Windows Settings' | 'Folder Redirection'.

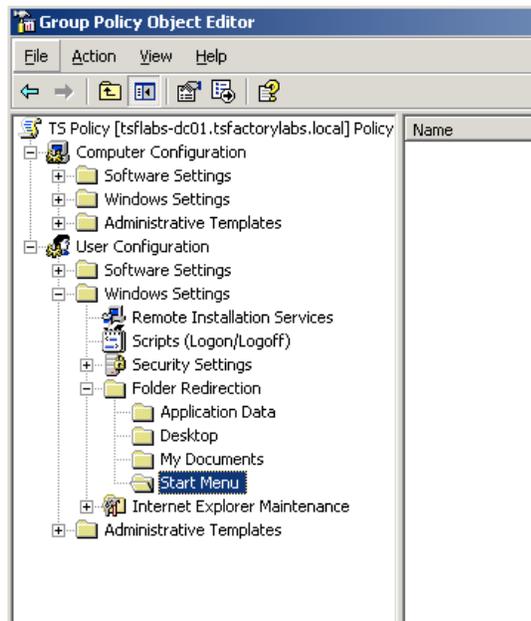


Fig. 53
Folder Redirection

3. Right Click 'Start Menu' and select 'Properties'. On the 'Target' tab select 'Advanced – Specify locations for various user groups' and click 'Add'.

Click 'Browse' to select the 'TS Users' group and then enter the path to the StartMenu\$ share as [\\your file server\StartMenu\\$](#). Click 'Ok'.

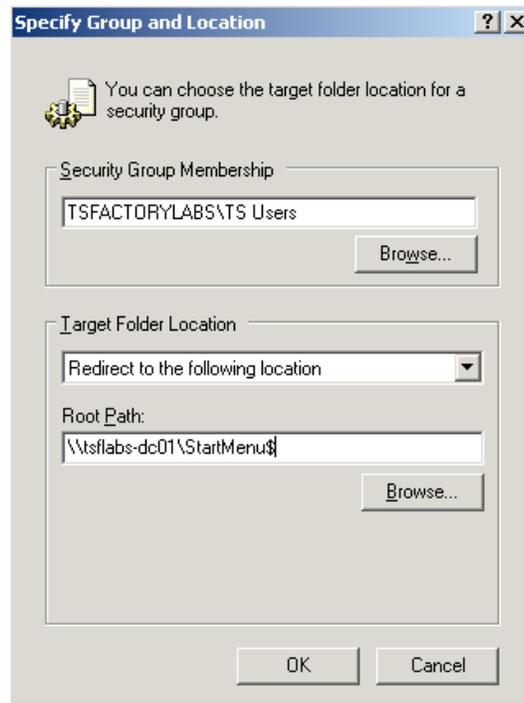


Fig. 54
Setting Group/Path for Start Menu redirection

4. If you did everything properly you should see a similar screen to the one below.

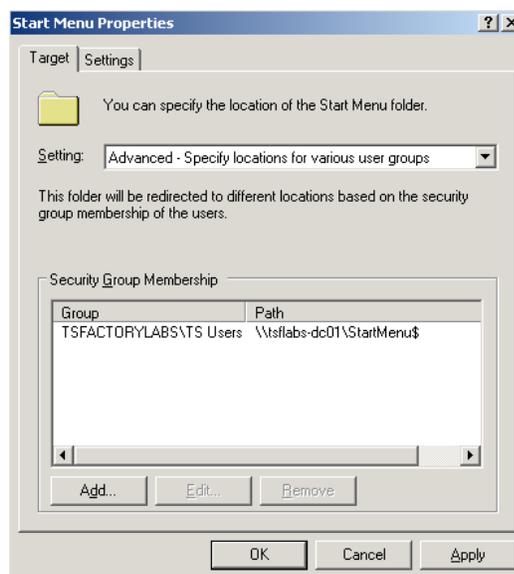


Fig. 55
The Start Menu redirection path for our TS Users group

5. Now repeat the same process for the 'Desktop'.

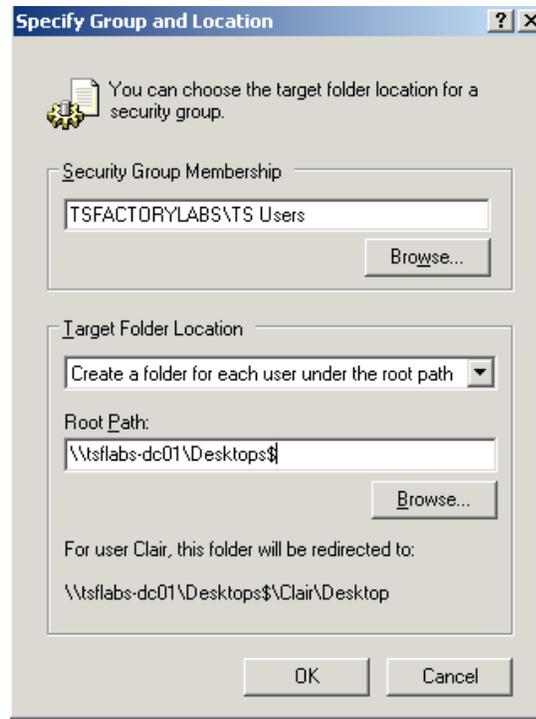


Fig. 56
Setting Group/Path for Desktop redirection

6. If you got it right, you should see something similar as below.

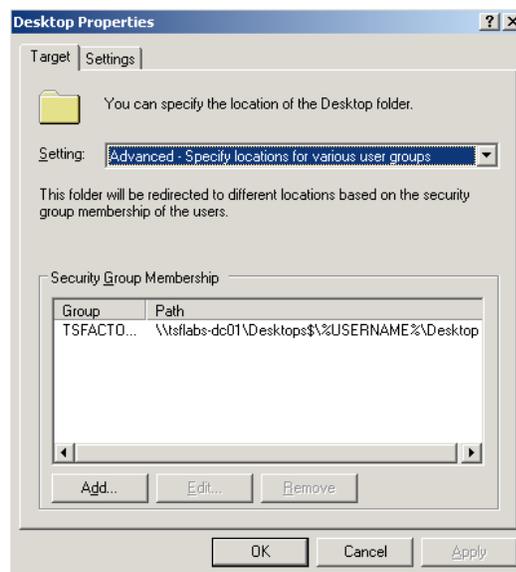


Fig. 57
The Desktop redirection path for our TS Users group

7. Just close 'Group Policy Object Editor' and you are almost set! The last thing to do is to pre-populate the Start Menu folder with the contents you want AND to set a small setting on our 'TS Policy' policy.
8. Logon to the TS with an Administrative account and right-click the 'Start' button. Select 'Explore All Users'. You will see the local Start Menu on the TS that is shown by default to all users. Right-click the 'Programs' folder and select 'Copy'.

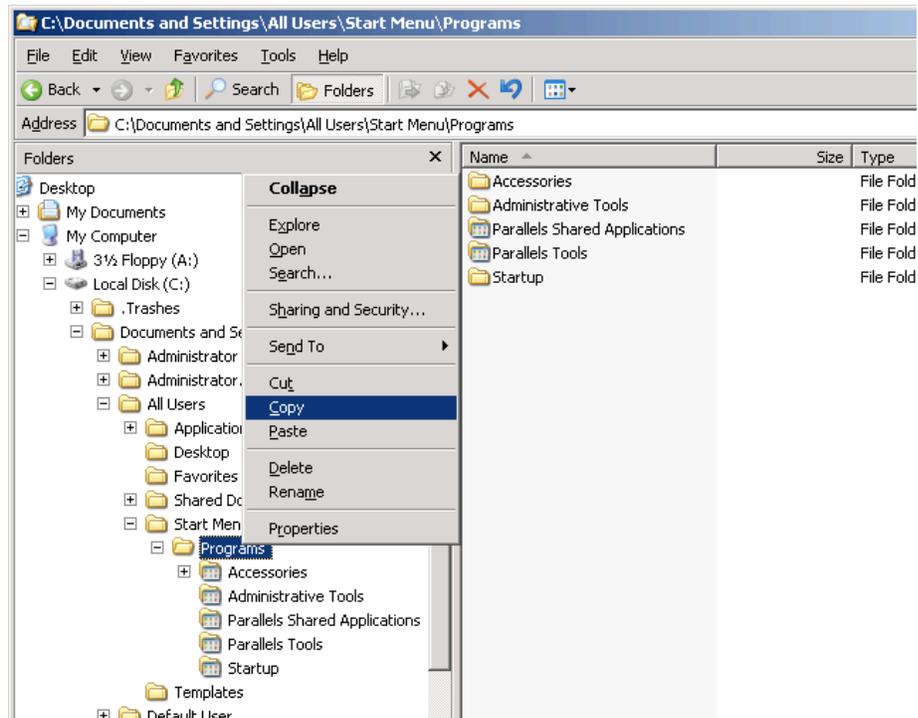


Fig. 58
Copying the default 'Programs' folder

9. Navigate to the 'StartMenu\$' share and paste the contents there.
10. Now simply navigate the 'Programs' folder you just pasted and remove the shortcuts/folders you do not want your users to see.
11. Finally launch 'Active Directory Users and Computers' and open our 'TS Policy' policy.
12. Navigate to 'User Configuration' | 'Administrative Templates' | 'Start Menu and Taskbar' and set the following policies:
 - a. Remove user's folders from the Start Menu: Enabled.

b. Remove common program groups from Start Menu: Enabled.

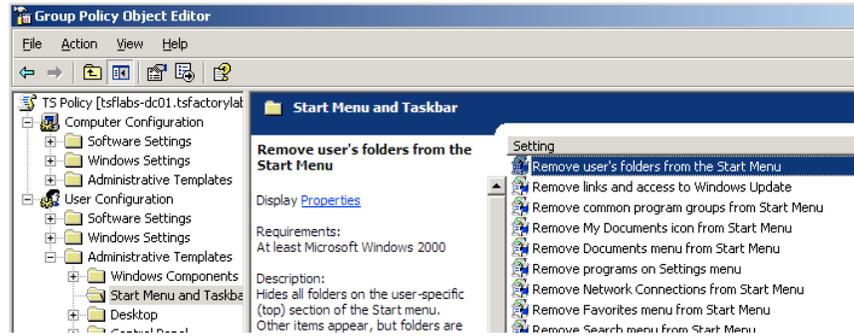


Fig. 59
Touching up the 'TS Policy' policy

13. Close 'Group Policy Object Editor' and reboot your TS or do a 'gpupdate / force' on it.

If everything was setup properly, when your users logon they will see a similar desktop to this!

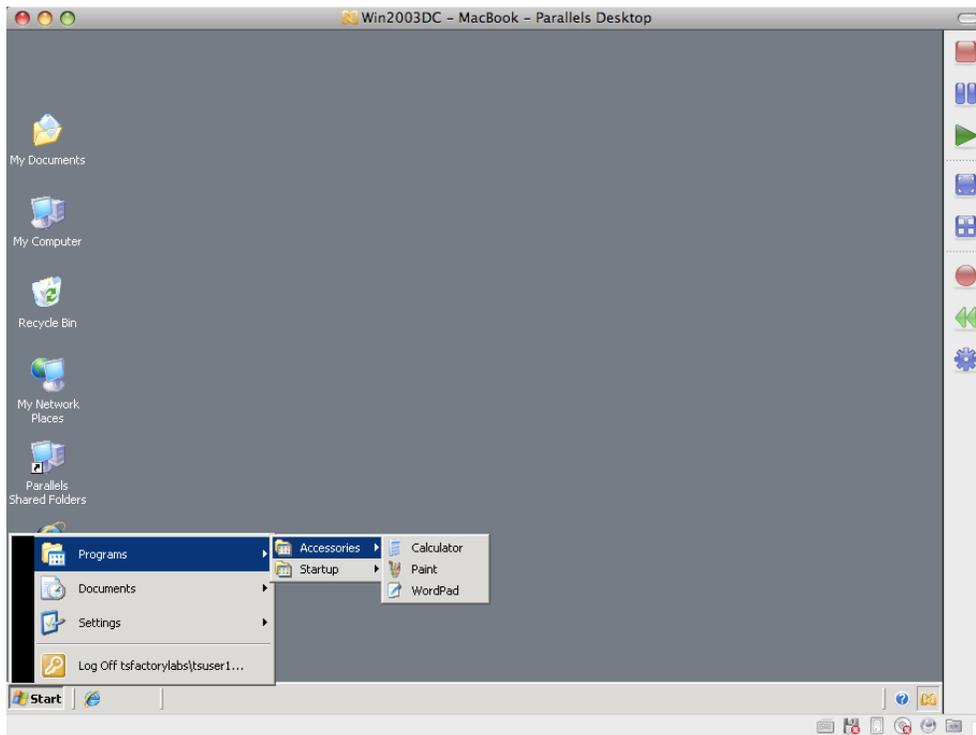


Fig. 60
A locked down desktop with 'Start Menu' redirection

Additional lockdown

With everything we have showed you so far you now have a much better terminal services environment! But as with anything else, there is always ways to improve it. Let's take a look at some of the additional steps you can take to further lock down your TSs.

SRP

SRP or 'Software Restriction Policies' is exactly what the name says: a policy to determine which applications (and types of applications) your users can and cannot launch. Although not perfect, SRP does increase the security on your TSs as users are now restricted to launch only the executables you authorize them to do! Several settings are available but it is not the intent of this guide to explain every single option available and the PROs and CONs on each one. There are already very good books/articles out there for this specific need.

But of course I will show you how to create a basic SRP for your TS. Feel free to expand it according to your needs!

1. Launch 'Active Directory Users and Computers' and open our 'TS Policy'.
2. Navigate to 'User Configuration' | 'Windows Settings' | 'Security Settings' | 'Software Restriction Policies'.

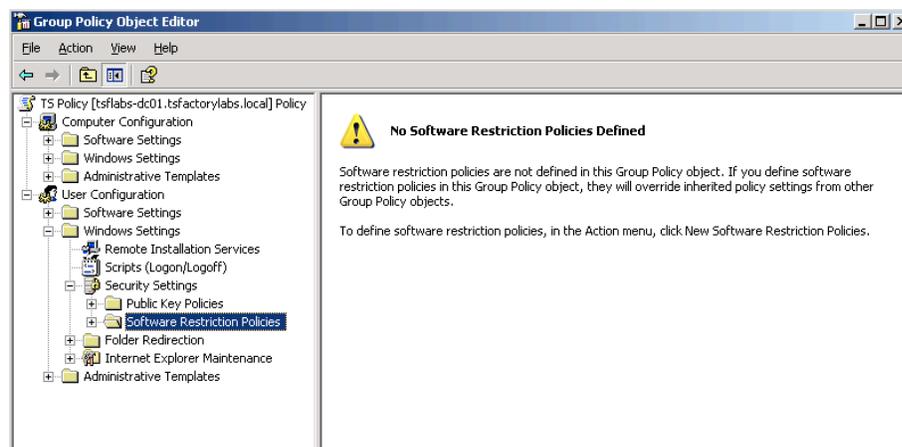


Fig. 61
Software Restriction Policies

-
3. Right-click 'Software Restriction Policies' and select 'New Software Restriction Policies'.
4. In this example we will deny access to Notepad. To do this right-click 'Additional Rules' and select 'New Path Rule...'. Now enter the path for Notepad.exe and type a description. By default this resource will be disallowed (exactly what we want). Click 'Ok'.

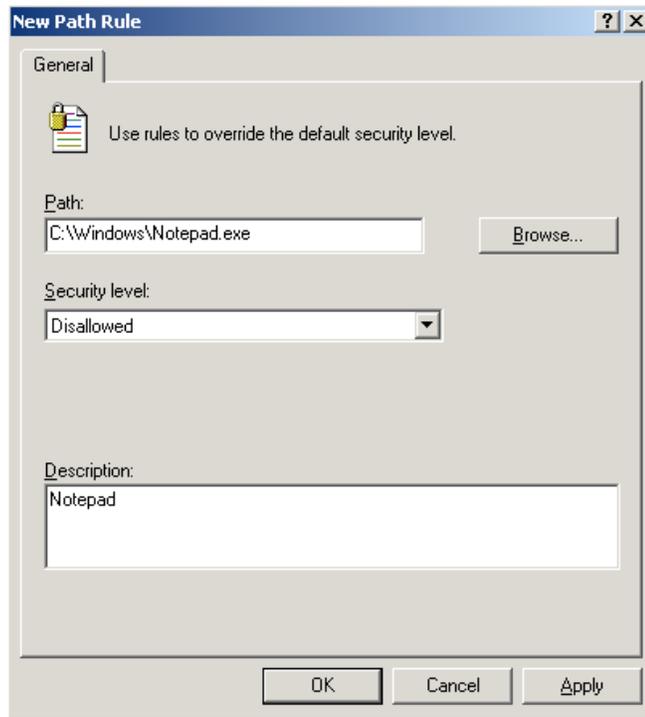


Fig. 62
Creating a 'New Path Rule' restriction

5. Your SRP screen should look like this:

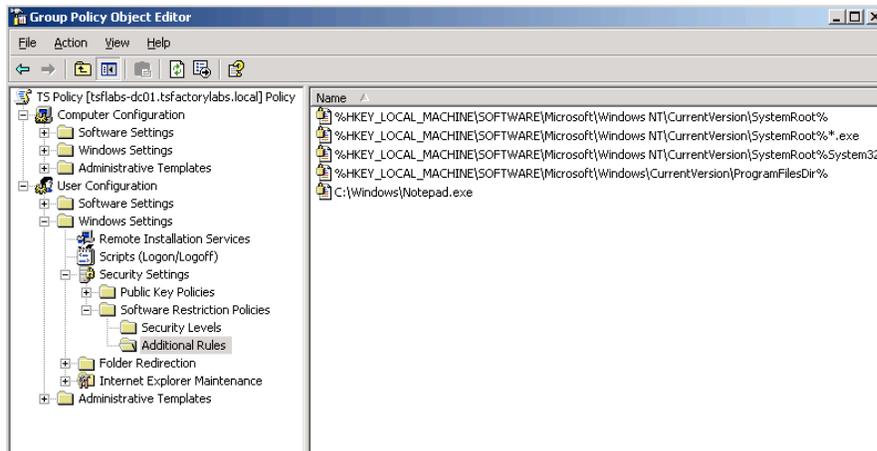


Fig. 63
Your SRP settings

6. Close 'Group Policy Object Editor' and reboot your TS or do a 'gpupdate / force' on it.

Now when your users try to launch Notepad they will see the following message:

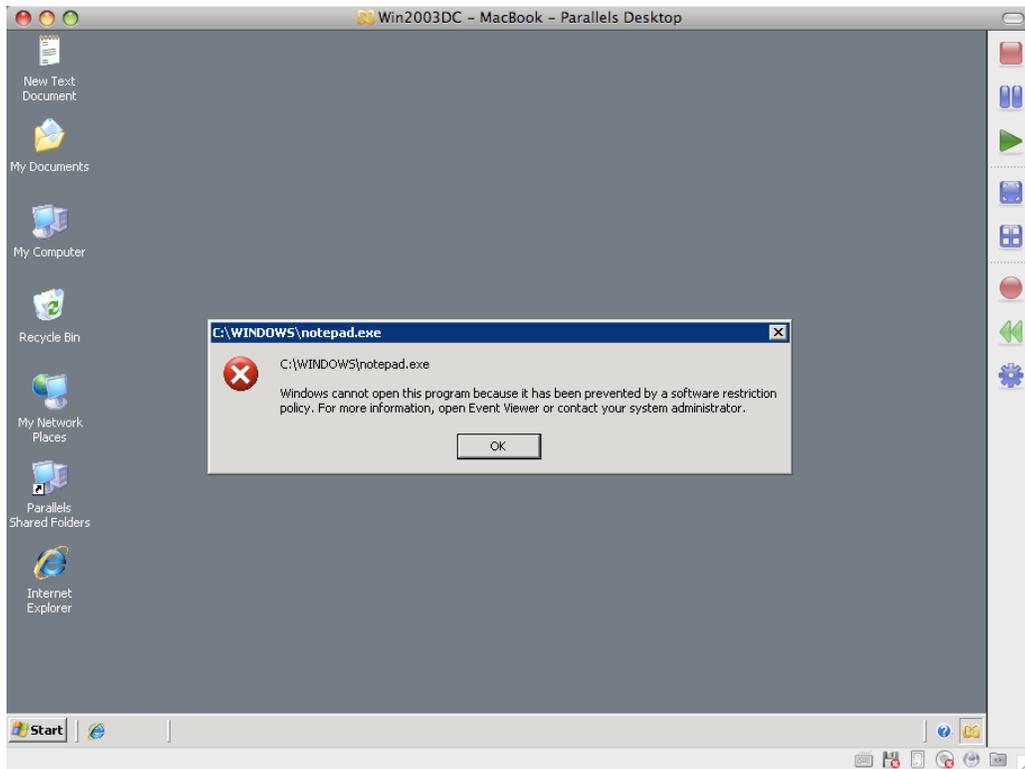


Fig. 64
SRP at Work

□

As you can see SRP can be pretty effective if used properly. Now do your homework and expand all the SRP stuff you just learned here.

SecureRDP

One of the most debated topics when talking about Terminal Services is how to access it. Some people say you should never expose port TCP 3389 to the Internet (the port TS uses for RDP traffic; we are not talking about exposing the whole server to the outside but just a single port); others, myself included, are not that paranoid. And in my case, I am still to see a case where a TS, that was properly configured, was hacked through RDP. So I am ok with the idea of having RDP exposed on the Internet, as long as the steps we discussed so far are in fact implemented and you add some extra security layer to the picture.

This is exactly where SecureRDP comes in. This is a small utility that we wrote back in 2004, extremely simple, lightweight and powerful. What SecureRDP does is simple: all connections coming to your TS are filtered based on criteria you set (i.e. username, computer name, client version, etc) and if the incoming connection matches the criteria the connection is allowed. If not, users receive an error message (fully customizable) before they can even see the logon screen and the connection is dropped.

One of my favorite filters is the client version number. If you read the article I wrote and posted on MSTerminalServices.org (Customizing the RDP client) you will learn you can change the RDP client to have a unique client version number (4-digit) and of course on SecureRDP you can filter by that. So if you create your custom RDP client with the version number only you know and deploy it to your clients, when you filter by this number, only your customized RDP client will be able to connect to your TSs. Simple and effective; and the best part, free.

All the details about this, from patching the RDP client to creating a new MSI file for it and the SecureRDP configuration are available at MSTerminalServices.org. Here you have the links:

<http://www.msternalservices.org/articles/Customizing-Microsoft-RDP-Client-Part1.html>

<http://www.msternalservices.org/articles/Customizing-Microsoft-RDP-Client-Part2.html>

External Access

Now that you have your environment up and running and properly locked down, how do you provide access to it from the outside? I could say this is actually very simple but as people have different requirements and concerns, for some this may become the hardest part of a TS implementation; for others, the easiest. Again, it all depends on how paranoid you are about security and your company policies/requirements.

All that is needed for TS to work (this includes everything users will need to do like printing, accessing their local drives, etc) is handled over port TCP 3389 (by default) by the RDP protocol. No other ports are needed or required.

As you can see the easiest way to provide external access to your clients is to simply configure your firewall to allow inbound connections on port TCP 3389 and redirect these to the TS internal IP address..

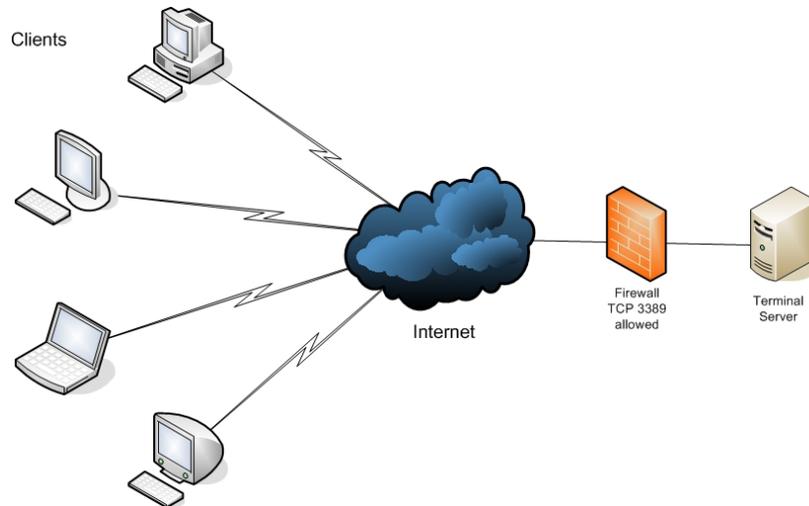


Fig. 65

□ External access to your TS – Port TCP 3389 only

i Note: I know we discussed this before but I think it is worth mentioning it again: TSWEB is not RDP over HTTP or HTTPS. If you setup TSWEB as per the steps described previously you still need port TCP 3389 opened and redirected to your TS. With TSWEB, as you can see, you will end up having two ports opened: TCP 80 or TCP 443 (http/https) to your web server where TSWEB is installed AND TCP 3389 (RDP) to your TS. Again, TSWEB is simply a mechanism to easily provide access to the RDP client itself for users that may not have it installed on their Windows machines.

Of course you can add some complexity to the picture to make it more secure. One typical example is to have a VPN in place so users need to connect first to your VPN service and then to the TS itself or you can have a two factor authentication solution (i.e. RSA SecurID) that will require users to enter their credentials with a token based number (that changes every minute!) to get access to the corporate network and then access your TS.

All these extra steps will add complexity to the users and may require several additional steps. And of course, if not properly configured, will cause more harm than good.

Scaling the environment

What happens next, based on my personal experience deploying TS all over the world is simple: users realize they can access pretty much any application using the TS thing, no matter where they are, and still with almost 'Office like' performance. So they start asking for more applications and more users start using the solution. Your one server TS solution must grow. So how do we do this?

Assuming all your applications coexist peacefully on the same TS all you need is to have more TSs exactly like your first one. Just get new servers and install them following the same steps you did for your first server. Of course this will work well if you have a two or three server TS environment; if you are dealing with ten, twenty or more servers your best alternative is to use an automated deployment tool like Altiris or visionapp (some tools are available at no cost when you purchase certain server brands and models – i.e. HP Blades). Regardless of the size of your environment, make sure you have up-to-date documentation of everything you do!

In case your applications conflict for some reason and cannot be installed on the same TS, you will need to have separate server groups, often called 'silos', each group with a different set of application.

Regardless of having a single or multiple silos, as you can see, it is just a matter of having multiple servers, all configured exactly like the other ones within the same silo. This leads us to the next topic. If you have multiple servers, how do you point your users to the servers? What if one server is not available? That is our next topic.

Load Balancing

With multiple servers available to your users, how do they know which server they should connect to? Ideally you want to make this process as transparent as possible to them. Something along the lines of "just connect to ts.ourcompany.com, no matter if you are here in the office or at home". For this to work you must implement something we call 'Load Balancing'.

'Load Balancing' is the process used to spread the 'load' (your users' connections) on your 'resource pool' (the TSs). There are several ways to do this and we will not be covering them here in detail for one simple reason: I wrote a two part article for MSTerminalServices.org called 'Load Balancing Terminal Services: all you wanted to know but were afraid to ask'; all you need to know is there, explained and with all the PROs and CONs for each alternative. Here you have the links:

<http://www.msterminalservices.org/articles/Load-Balancing-Terminal-Services-Part1.html>

<http://www.msterminalservices.org/articles/Load-Balancing-Terminal-Services-Part2.html>

But as I do like my readers let's take a quick look at what you should look for when trying to find a solution to load balance your TSs and the problems you may experience.

First of all, when users try to connect to your TSs, the best option would be to have some intelligent mechanism that would determine the best TS available at the moment based on how busy all your TSs are. This mechanism would also determine if a TS is actually responding. This is what we call a 'Resource Based Load Balancing' and we can easily see the benefits associated with it: users always get the best performing TS (as they are never directed to a TS that is at 100% CPU for example) and if a TS is not available, they do not even know it. So keep this term in mind when looking for a load balancing solution: 'Resource Based'.

The second thing you must address was taken care of at the beginning of this guide. When users logon to a TS to run their applications, certain settings may be initialized and saved for that particular user (i.e. his Microsoft Outlook mailbox settings and preferences) on the TS he is connected to. But when the user logs off and connects back later, he may now be on a different TS (assuming you have a load balanced environment). In this case you want his/her settings to 'follow' him/her; regardless of the TS he/she is connected to. This is done by using roaming profiles that we set at the beginning. So now you understand why roaming profiles (or a profile solution that 'follows' the user) is required!

The final issue you must be aware is called 'reconnection'. Let's say a user is connected to a TS when all of a sudden his/her connection drops (i.e. his ISP was down for a couple minutes). When he tries to reconnect, he expects to be reconnected back to his existing session that was running on that particular TS and not to get a new session running on a less busy TS. So your load balancing mechanism must be intelligent enough to know at any time where each user has his session so it can reconnect them in case the connection drops.

I must emphasize that each environment is different meaning that needs and requirements are different as well. For some the old Microsoft Network Load Balancing will suffice, even though it is not resource based and does not know how to reconnect users; for others resource based load balancing and reconnection capabilities are mandatory requirements. Determine what your needs are, read the articles I mentioned and then find the best solution.

As of today, one of the most impressive and well-proven load balancing solutions out there is 2X LoadBalancer. It is not only resource based, but it also knows how to deal with reconnections, in case a user session gets dropped. This level of functionality is usually found on hardware load balancers or other software solutions costing thousands and thousands of dollars more. With many customers running the 2X LoadBalancer worldwide, it is the most reliable and accessible solution you will find.

Scalability

By scalability I mean how to determine the number of users a server can handle so in case you are adding more users to your TSs, you know before hand how many TSs you may need to add.

There are several ways to determine this. Some will be cheap (meaning free) while others may cost something (usually meaning extremely expensive). Again, it is all about what you need and how much you are willing to spend.

The cheapest way is to use our old and well known Performance Monitor (perfmon), installed out-of-the-box with any Windows Server out there. By simply monitoring a server (and collecting all this data for further analysis) you will be able to determine at which user load performance may become unacceptable (yes, I know this is very subjective – some users, no matter what you give to them, will always say everything is slow). For example, if you determine this number to be 70 users, plan your environment to have a maximum of 80% of this user number per server (in this example, 56 users) so in case one server goes down for some reason (maintenance, network issue, etc) there is still room on the remaining servers to handle the extra load required for the users that were using the server that is not available anymore.

In case you prefer to simulate this load on your TSs to determine the ideal number of users per box, you can use tools like 'AutoIT' (freeware) or fancier ones (i.e. LoadRunner, Citraset VU, EdgeSight for Load Testing, etc).

Bandwidth Considerations

I know I mentioned before the most asked question on the Internet related to Terminal Services is printing. Right after printing, the second one on the list is about bandwidth. Everyone wants to know how much bandwidth will be needed if they have an X amount of users connected to a TS.

Honestly, there is no way to give you an answer for this question. It all depends on the applications your users will be using, if they print like crazy, if they will be listening to internet radio through your TSs and so on!

Although I cannot give you a number, I can tell you how to find out the number you are looking for and also give you some ideas on what can be done to improve the overall perceived performance your users will have when using your TSs.

The first step is to determine how much bandwidth a typical user will need. To do that you can use the same Performance Monitor mentioned above or some sniffer (a tool that will capture all packets going through a network port and/or device) like Wireshark (free!) while a typical user uses a separate TS for a day.

This will show you the total bandwidth for the time he was connected and its highs and lows.

To preserve the experience your users need when on the TS you should consider a couple things: RDP, as any other protocol, needs bandwidth and the less bandwidth you have, the slower things will move. Therefore, if you want to guarantee a certain performance level for your users, you must have a way to guarantee that RDP bandwidth will never go lower than a certain threshold. This number could be the average bandwidth needed by a typical user that you determined using perfmon or a sniffer as mentioned above. To achieve this you will need network devices that can control how much bandwidth a protocol may use. With such devices you can for example set the maximum amount web browsing will use (HTTP/HTTPS) so it does not affect how much bandwidth RDP has available. These devices can also guarantee a minimum number for each protocol (and a maximum). If your TSs are sharing the same network link with your mail servers, internet browsing, etc, it is clear to see why this becomes important!

I am not saying that you will necessarily need to have traffic prioritization and/or bandwidth controls in place.

Check your connections and how they are being used (wireshark for example can show you the total amount of data used per protocol) before you blame Terminal Services.

Enhancing the environment

Now that you have a TS environment up and running and with multiple servers load balanced (I assume you got the 2X LoadBalancer), what else can we do to enhance the overall end-user experience? Actually there are probably many things that could be done but I will just list a few of them.

Seamless Windows

For those not familiar with this very common term in the TS world, this is very easy to understand and as always, a picture is worth a thousand words! So let's take a quick look at the following screenshots.

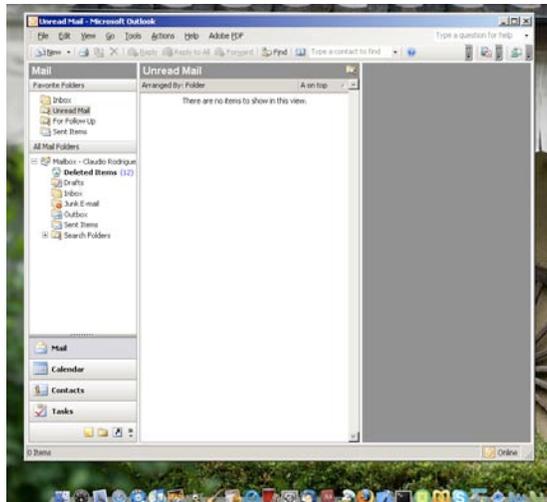


Fig. 66
Seamless Windows Application

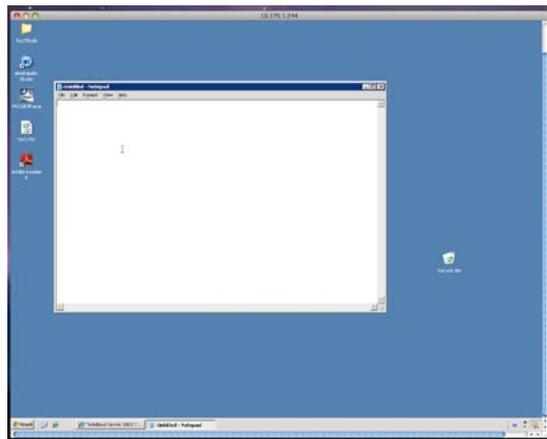


Fig. 67
Non-seamless Windows Application

As you can see, in the first screenshot Microsoft Outlook seems to be running 'locally'; there is no window frame around it. On the other hand, on the second screenshot, Notepad is running but you can see the TS start button and task bar, and the window frame (with the close, minimize and restore buttons on it) around it. Not as 'seamless' as the first example where the application actually seems to be running locally on my computer!

The ability to give individual applications to the users instead of a 'Full desktop window' is called application publishing with seamless windows. The 2X ApplicationServer product allows you to 'publish' individual applications that will look and behave as if they were locally installed on the user PC. This is not only much cleaner to the user (as they do not have 'two taskbars' anymore!) but a much better way to introduce your users to the benefits of Server Based Computing. All this at very little cost.

Firewall Friendly Access

In many cases you may need to provide your users (or partners for that matter) access to your TS infrastructure over the Internet and as we know, sometimes these users/partners will be connecting from networks where you have little or no control whatsoever. The typical end result of that is port TCP 3389 being completely blocked and in more restricted places, only HTTP or HTTPS traffic will be allowed. This means even if you change the TS listening port (yes, you can do that) to something else like 21 (trying to trick the firewall that this may be FTP traffic.) it will not work.

So what is the solution? Well the best way is to encapsulate (or tunnel) RDP traffic over HTTPS. This basically establishes an HTTPS connection from the client to the TS.

Good to know but what should I use? One of the first and now most mature products in the market is the 2X LoadBalancer. It not only load balances your TSs in an intelligent manner but also provides a complete RDP over HTTPS solution for your clients! When used with the 2X ApplicationServer, users will be able to access individual applications ('published applications') over HTTPS, making the solution completely firewall friendly.

Conclusion

You probably understand more about Terminal Services than when you started reading!

One more time, all I wanted when writing this guide was to give people out there a better understanding of Terminal Services (and Server Based Computing) in general, showing its strengths and weaknesses but without getting into the technical details and low level tweaks that you may need one day. This guide was written with the TS beginner in mind, trying to give them the basics required for a successful TS deployment. And I hope that is exactly what you got when you finished reading this guide.

I would also like to thank Niko Makris at 2X Software Ltd.; Niko and his team are the main reason why this guide became a reality, always encouraging me and dealing with the inevitable delays that come with writing guides.

And in case you have comments, suggestions, rants, whatever about this guide, feel free to contact me at any time. Just look for TSMVP on Experts-Exchange.com.